The Dissertation Committee for Duo Ding
certifies that this is the approved version of the following dissertation:

CAD for Nanolithography and Nanophotonics

Committee:

David Z. Pan, Supervisor

Ray T. Chen

Joydeep Ghosh

Michael E. Orshansky

J. Andres Torres

Nur Touba

CAD for Nanolithography and Nanophotonics

by

Duo Ding, B.S.; M.S.E.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2011

Dedicated to my parents:

DING Bin and HAN Dongwen

for their unfailing love.

To my soulmate Wing-Chi Debbie Lee,

Wencui Clara Chen and my dear brothers and sisters

in the big family of TCA

without whom this work

could never have been possible.

Acknowledgments

I have been very fortunate and deeply honored to have Professor David Z. Pan as my Ph.D. advisor during my graduate study at the University of Texas at Austin. He is not only a great mentor to me in my academic career but also a wise and dear friend who teaches me to pick up my courage again during the gloomy days of life. I am deeply indebted to him since none of these would have been possible without his trust and support.

I express deep gratitude and appreciation to the members of my Ph.D. committee for the precious time and efforts made out of their busy schedules. In particular, I would like to thank Professor Michael Orshansky for his support and technical suggestions during the development of this dissertation. I would like to thank Professor Ray T. Chen for all the opportunities and supports that he generously offered during the years. I would like to thank Professor Joydeep Ghosh for his patience and many valuable suggestions during the development of this dissertation. I would like to thank Professor Nur Touba for his kindness and professional insights to the development of this dissertation. Finally I would like to thank Dr. J. Andres Torres for being a great mentor and a dear friend during my internship at Mentor Graphics Corp., which has deeply impacted my career development.

My gratitude and appreciation also go to my colleagues at the Univer-

sity of Texas at Austin Design Automation Group for their great support and generous sharing: James Ban, Ashutosh Chakraborty, Minsik Cho, Jhih-Rong Gao, Ou He, Wooyoung Jang, Kiwoon Kim, Anurag Kumar, Yen-Hung Lin, Katrina Lu, Joydeep Mitra, Jiwoo Pak, Xiaokang Shi, Jae-Seok Yang, Kun Yuan, Bei Yu, Wen Zhang and Yilin Zhang.

I dedicate my dissertation to my parents in China, my father *Bin* and my mother *Dongwen*, without whose prayers, love, encouragement and sacrifice I would not have been half a man as I am. My sincere thanks to my uncle *Jianwen*, auntie *Jing* and brother *Joseph*. I am forever indebted to their love and support throughout the years. I express my deepest gratitude to Wing-Chi Debbie Lee, Wencui Clara Chen and the clouds of brothers and sisters in the big family of TCA, where I have learned to think, to love, to give and to be thankful.

Portions of this work were supported by: SRC, NSF Career Award, Texas Advanced Research Program, supports and equipment donations from IBM, Intel, Oracle and Fujitsu.

CAD for Nanolithography and Nanophotonics

Publication No. _____

Duo Ding, Ph.D.
The University of Texas at Austin, 2011

Supervisor: David Z. Pan

As the semiconductor technology roadmap further extends, the development of next generation silicon systems becomes critically challenged. On the one hand, design and manufacturing closures become much more difficult due to the widening gap between the increasing integration density and the limited manufacturing capability. As a result, manufacturability issues become more and more critically challenged in the design of reliable silicon systems. On the other hand, the continuous scaling of feature size imposes critical issues on traditional interconnect materials (Cu/Low-K dielectrics) due to power, delay and bandwidth concerns. As a result, multiple classes of new materials are under research and development for future generation technologies.

In this dissertation, we investigate several critical Computer-Aided Design (CAD) challenges under advanced nanolithography and nanophotonics technologies. In addressing these challenges, we propose systematic CAD methodologies and optimization techniques to assist the design of high-yield and high-performance integrated circuits (IC) with low power consumption.

In Very Large Scale Integration (VLSI) CAD for nanolithography, we study the manufacturing variability under resolution enhancement techniques (RETs) and explore two important topics: (1) fast and high fidelity lithography hotspot detection; (2) generic and efficient manufacturability aware physical design. For the first topic, we propose a number of CAD optimization and integration techniques to achieve the following goals in detecting lithography hotspots: (a) high hotspot detection accuracy; (b) low false-positive rate (hotspot false-alarms); (c) good capability to trade-off between detection accuracy and false-alarms; (d) fast CPU run-time; and (e) excellent layout coverage and computation scalability as design gets more complex. For the second topic, we explore the routing stage by incorporating post-RET manufacturability models into the mathematical formulation of a detailed router to achieve: (a) significantly reduced lithography-unfriendly patterns; (b) small CPU run-time overhead; and (c) formulation generality and compatibility to all types of RETs and evoling manufacturing conditions.

In VLSI CAD for nanophotonics, we focus on three topics: (1) characterization and evaluation of standard on-chip nanophotonics devices; (2) low power planar routing for on-chip opto-electrically interconnected systems; (3) power-efficient and thermal-reliable design of nanophotonics Wavelength Division Multiplexing for ultra-high bandwidth on-chip communication.

With simulations and experiments, we demonstrate the critical role and effectiveness of Computer-Aided Design techniques as the semiconductor industry marches forward in the deeper sub-micron ($45nm$ and below) domain.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

As the semiconductor industry marches towards deeper sub-micron scale of feature sizes, the traditional VLSI circuit design and manufacturing cycles have become more and more critically challenged. In this dissertation, we study and explore two important emerging technologies, namely the nanolithography and the nanophotonics, to assist the optimization of two most critical aspects for modern VLSI ICs, i.e., manufacturing yield and interconnect performance.

1.1 VLSI CAD for Advanced Nanolithography

With rapid advances of semiconductor process technology [8], the minimum feature size of modern ICs is becoming smaller and smaller than the lithographic wavelength. In order to bridge such a wide gap between high integration demands and manufacturing limitations, the *Manufacturability Aware Design* paradigm is introduced and applied to create highly reliable IC's under the scaling-down feature size [3, 27, 34, 89, 103, 141]. Known as *Design for Manufacturability* (DFM) techniques, these methods ensure high manufacturing yield by incorporating manufacturability models into early design stages

to avoid lithography-unfriendly patterns (usually referred to as lithography hotspots).

For modern VLSI circuits design, a typical DFM flow consists of a physical verification stage which identifies the lithography hotspots followed by a series of techniques either to fix these hotspots in a construct-by-correction manner or to avoid them in a correct-by-construction manner. Either way it is, fast and high fidelity hotspot identifiers serve very critical roles for high yield IC design. On the one hand, approaches that employ lithographic simulations [67, 103] are precise yet costly to run due to the time-consuming calculations involved. On the other hand, approaches that utilize pattern/graph matching techniques [66, 135, 138] may suffer from high false alarms (upto over 60 times false alarms than actual hotspots [66]) and other issues especially when pattern enumeration is too costly to perform a priori. Moreover hotspot patterns are hard to apply - too many patterns lead to high over-estimate rate (false-alarms) and too few patterns result in low detection accuracy.

In the meantime, there are works that incorporate modern data mining methods to achieve fast and accurate hotspot detection. A neural network judgment detection flow was proposed in [92], where hotspot image patterns were used for training a compact Artificial Neural Network (ANN) model. Also in [83], data mining algorithms are developed for hotspot pattern clustering. While these early attempts have shown promising potentials, there are still many limitations to overcome, such as high training noise and low detection accuracy, etc.

In face of these challenges, a new set of hotspot detection methods need to be developed to combine all the strengths of the above methods. Such new methods will greatly leverage DFM techniques not only by detecting hotspots at-speed but also helping prevent high variability (poor manufacturability) patterns in the early design stages.

In Chapter 2 of this dissertation, we aim to develop such techniques for high performance lithography hotspot detection then apply them in the early physical design stages to improve IC manufacturability and yield. We first start with Section 2.1 to briefly overview the background of nanolithography technology and the basics of resolution enhancement techniques (RETs).

In Section 2.2 we introduce the concept of critical hotspot feature as a compact representation of the original pixel based design layouts. Compared with 2D pixel images used in [83, 92], critical features effectively reduce the dimension of the original raw data and filter out detection noise. With powerful machine learning algorithms/models [50, 118] trained with the critical features, our method demonstrates high hotspot detection speed with good accuracies and small false-alarms (10% of actual hotspots).

Based on such a method, in Section 2.3 we propose techniques for further accuracy improvement and false-alarm suppression using multiple machine learning models and successive detection refinements. Industry-strength benchmarks and advanced RETs are employed in the simulation experiments under the real manufacturing conditions. The results demonstrate satisfactory overall performance in terms of detection accuracies, false-alarms and

run-time.

In Section 2.4 we push further to propose a generic and unified meta-classification framework to combine the strengths of various disparate hotspot detection techniques. In this section, various machine learning techniques and pattern matching methods are developed and experimented under the proposed framework, which achieves very impressive capability and overall performance trade-off between hotspot detection accuracies and false-alarms.

In Section 2.5 we explore the applications of hotspot detection methods in the early design stages to assist correct-by-construction design flows. In this section, we develop a fast and generic formulation for a manufacturability-friendly detailed router. Our formulation outperforms previous state-of-the-art lithography-friendly detailed routers at small CPU run-time cost.

1.2 VLSI CAD for On-chip Silicon Nanophotonics

As raised in the International Technology Roadmap for Semiconductors [8], silicon system complexity rockets exponentially due to increasing transistor counts, fueled by smaller feature sizes and increasing demands for higher integration densities. Consequently, interconnect design becomes more and more important for deep sub-micron (DSM) VLSI circuits. Among various types of interconnect technologies, nanophotonics is becoming a potential quantum leap towards next-generation on-chip interconnect. Ever since its introduction [51], the concept of on-chip optical interconnect has attracted more and more attention over the years in the industry (e.g., [68, 123]) and academia

(e.g., [25, 86, 94]) especially at device fabrication level. As projected [22], on-chip optical interconnect outperforms traditional copper interconnect in power, throughput and delay at below 22nm technology nodes starting from around 2016.

In the recent years, manufacturing technology has evolved to make on-chip nanophotonic devices using silicon-on-insulator process. Consequently, nanophotonic devices could be manufactured in compatibility with the existing CMOS devices. These silicon compatible nanophotonic devices (e.g., [56, 123]) are built for optically interconnected ICs with advantages including but not limited to: fast signal speed, ultra low power and very high on-chip bandwidth compared with traditional Cu/low-K interconnect. New advances in silicon nanophotonics devices - such as photonic crystal structures [55, 125] - have also been demonstrated with rigorous practices. In the recent years, low RF power optical modulators operating at a few Gbps speed have been developed [53, 55] with compact footprints for potentially large density integration. Compact photodetecters with up to 50Gbps speed are also demonstrated, such as Germanium-on-Insulator photodetecters [69].

Most of the above work focus on optimizing the performance and power of individual devices. Few works have been proposed to study the integration challenges and solutions at VLSI circuits and physical design level. With a carefully characterized standard library of the basic nanophotonic devices, we will be able to perform device simulations and explore the design space of opto-electrically interconnected integrated circuits.

As a previous work, [87] studied timing driven and congestion driven on-chip optical routing CAD algorithms for 3-D system-on-package. Yet the routing geometry in [87] was formulated in a simple manner: point-to-point straight connection with at least 1 optical modulator inserted at each pin and Steiner node of the circuit netlist. There are 3 major issues. *First*, it neglects the laser power consumption of optical modulators. Since each modulator requires a laser source for electrical-to-optical data conversion, this approach results in a very power hungry design. *Second*, it neglects the photon-energy loss on the optical interconnect. Consequently, there could be pins whose received photon-energy drop below the photo-detection threshold, leading to logic errors after optical-to-electrical data conversions. *Third*, optical routing has very different characteristics compared with conventional electrical (Cu) interconnect routing, therefore special routing geometry must be developed to tackle optical interconnect planning problems. In other words, total laser power consumption (proportional to number of modulators inserted) and the constraints for successful optical-to-electrical detection must both be addressed properly for the optimized optical routing solutions.

In view of these issues, in Chapter 3 we will build a standard optical device library and study various CAD techniques to address the high performance low power integration challenges in the cross-domain of nanophotonics and electronics.

In Section 3.1 we start with the introduction of an Optical Interconnect Library: OIL for the modeling of current and near future nanophotonic devices

as basic building blocks for later sections.

In Section 3.2 we propose a low power routing framework to take into considerations of various physical constraints and new design characteristics in the optical domain. Such a framework is constructed based on OIL library of Section 3.1.

In Section 3.3 we further examine the thermal-reliability issues of the nanophotonics devices and propose a CAD flow to achieve low-power thermal-reliable circuit integration utilizing on-chip optical Wavelength Division Multiplexing devices.

We will summarize and conclude this dissertation in Chapter 4.

Chapter 2

VLSI CAD for Advanced Nanolithography

2.1 Nanolithography and Resolution Enhancement

In this section we first briefly overview the nanolithography technology and some state-of-the-art Resolution Enhancement Techniques (RETs).

2.1.1 Introduction to Lithography Imaging Systems

As modern DSM VLSI advances, the widening gap between the design feature sizes and the lithography wavelength (depicted in Fig. 2.1) creates numerous critical challenges for designing next generation VLSI circuits and systems. Before addressing these challenges, we first introduce some preliminaries of the current main-stream nanolithography system.

Shown in Fig. 2.2, the optical lithography system consists of the light source, the mask, the objective lens and the wafter, where the high energy laser source sheds light on the mask and exposes the wafer through a set of objective lens. We define the Critical Dimension (CD) of the lithography imaging system as the half pitch size written as follows in Equation 2.1,

$$CD = k_1 \cdot \frac{\lambda}{NA} \tag{2.1}$$

where λ is the wavelength of the laser source (currently 193nm), NA is the

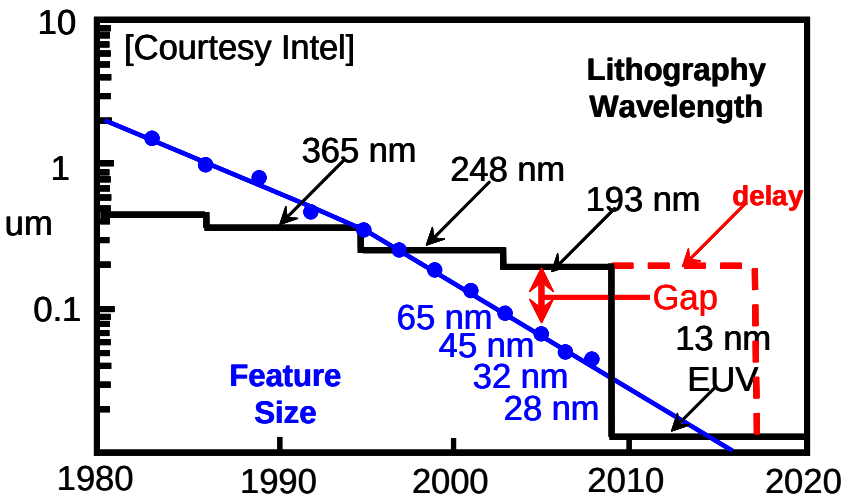
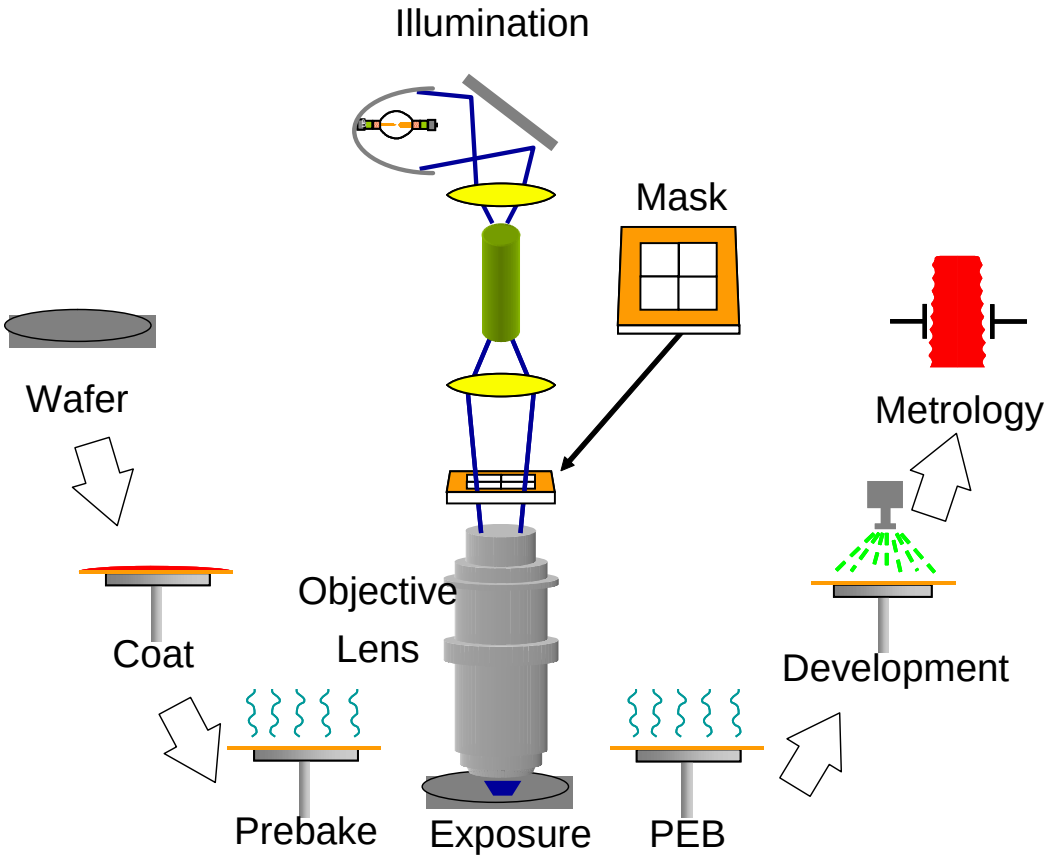


Figure 2.1: Gap between shrinking feature sizes and lithography wavelength

numeric aperture of the optical focal system, k_1 is a process-related parameter. For smaller feature sizes, we need smaller CD and larger NA.

Unfortunately the half pitch size of the current optical lithography system is reaching its fundamental limit ($k \sim 0.25$) and severe variations are observed on the wafer at sub-45nm technology nodes. They come from various sources such as topography variation, focus/dosage variation, mask size variation and CD variation. They cause critical issues to design closure and manufacturing closure in timing, power and yield. Sometimes these variations create reliability issues even after the chips have been manufactured, especially under temperature variation, strain/stress silicon effects and electron migration effect, etc. Before the next generation Extreme-Ultra Violet (EUV) Lithography reaches mature high volume production, these variations will continue to dominate the deeper sub-micron (45nm and below) domain.



CD is the **minimum feature size** that a projection system can print:

$$CD = k_1 \cdot \frac{\lambda}{NA}$$

k1 is process-relate
NA is numerical aperture
λis currently stuck at 193nm

Manufacturing process **NOT** perfect:
Topography variation
Focus variation
Dosage variation
Mask size variation
LER variation
CD variation
Etc.

Figure 2.2: An illustration of the nanolithography imaging system

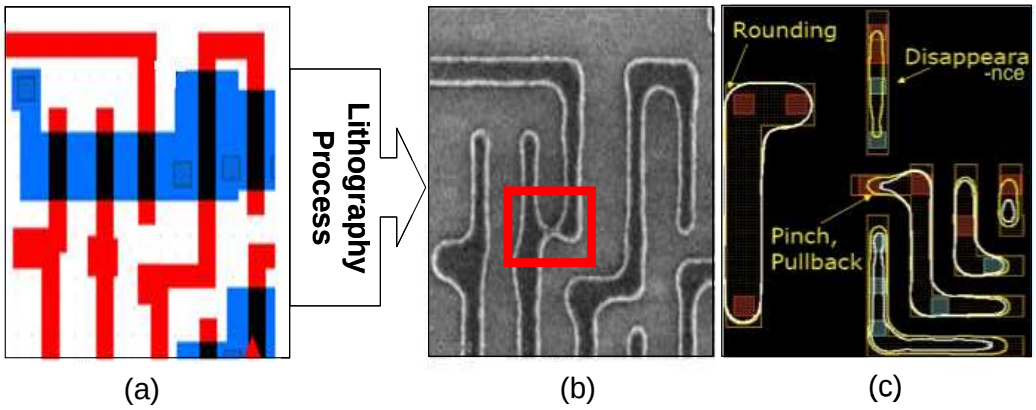


Figure 2.3: A few examples of lithography (process) hotspots

2.1.2 What We See Is NOT What We Get

Due to the Critical Dimension limitations of the imaging system, deep sub-micron VLSI circuits suffer from many types of manufacturing variations at different degree levels. At sub-65nm technology nodes, such variations are no long negligible since they introduce both functional and electrical errors to the silicon wafter. In other words, what we see at the design stage (Fig. 2.3(a)) is not what we get at post silicon stage (Fig. 2.3(b)).

To capture such an effect, accurate lithographic simulators (involving Hopkins equations and computational lithography) have been rigorously developed and employed to model the real manufacturing conditions and re-target designs for better performance prior to tape-out. Accurate lithography simulations calculate the Process Variation (PV) Bands of a design layout under certain specified optical system models. Various types of hotspots located

by lithography simulators are shown in Fig. 2.3(c), where process variation has created silicon failures such as rounding-offs, disappearance, pinches and pullbacks on metal and poly layers.

2.1.3 Resolution Enhancement Techniques

To compensate the above effect, various RETs have been introduced to equivalently reduce the k factor of Equation 2.1 during the mask optimization. Current main-stream RETs include but are not limited to Sub-Resolution Assist Feature Insertion and Optical Proximity Correction (OPC) [60]. We illustrate the effectiveness of these techniques in Fig. 2.4, where the printability of design patterns are greatly enhanced as RETs are applied.

Fig. 2.4 starts with a double-T shape pattern in 227nm technology node and shrink down the pattern to 68nm node. Since the optical illumination uses 193nm wavelength lithography, the pattern printability becomes worse and worse until a functional error occur (shorting) at 68nm half-pitch dimension. By employing various RETs such as Off-Axis Illumination (OAI), biasing, scattering bar insertion and full-blown OPC, the original design pattern is significantly improved until its printability becomes similar to that of 227nm node. In other words, RETs are serving as powerful weapons in the technology scaling battle.

In recent years, more powerful RETs are under active research and development, such as Litho-Etch Litho-Etch Double Patterning Lithography [16] and Self-Aligned Double Patterning Lithography [10, 80], etc.

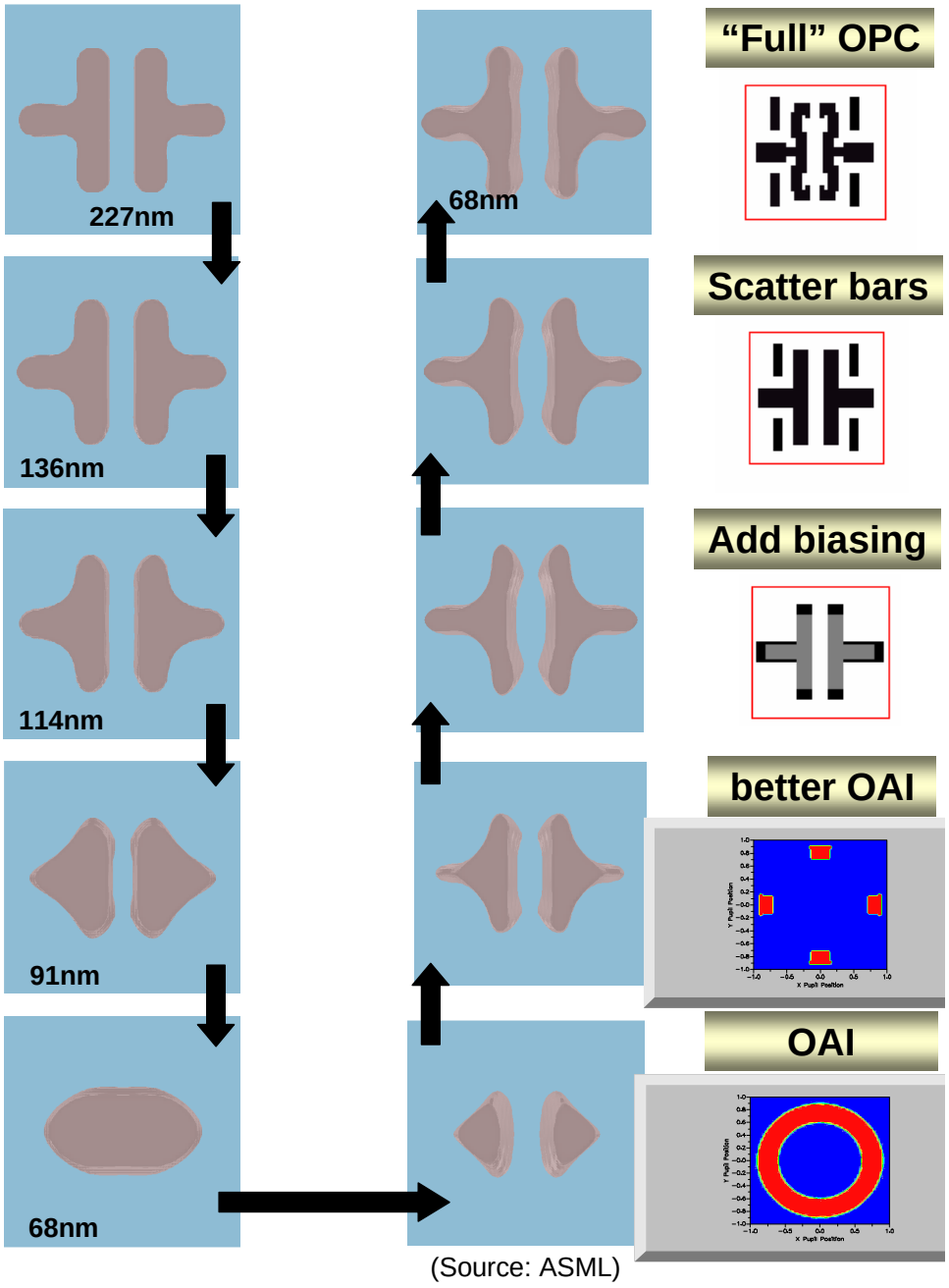


Figure 2.4: An example of different lithography imaging resolutions using various RETs for the same structure (Source ASML).

2.2 Lithography Hotspot Detection Using Signature Extraction and Machine Learning Classification

With rapid advances of semiconductor process technology [8], the minimum feature size of modern IC is becoming smaller and smaller than the actual lithographic wavelength. In order to bridge such a wide gap between design demands and manufacturing limitations, various popular *manufacturability aware design* techniques have been proposed and applied towards high fabrication yield and resilience for designs with scaling-down feature sizes [3, 27, 34, 89, 103, 141]. Successful *design for manufacturability* (DFM) techniques ensure high fabrication yield by incorporating manufacturability aware models into design stage to avoid potentially problematic patterns (usually referred to as process hotspots).

One typical DFM flow requires a physical verification stage for fast and high performance lithography hotspot detection followed by a series of steps to correct these hotspots. On the one hand, approaches that employ lithographic simulations [67, 103] are precise yet costly to run. On the other hand, pattern or graph matching techniques [66, 135, 138] could suffer greatly from high false-alarms or other issues since hotspot patterns are very hard to apply - too many patterns lead to high over-estimates and too few patterns result in low accuracies and coverage.

Recently there are works that start incorporating modern data mining methods towards fast and accurate hotspot detection tasks. A neural network judgment based detection flow was proposed in [92], where hotspot image

patterns were used for training an artificial neural network (ANN) kernel. Also in [83], data mining algorithms are developed for hotspot pattern clustering. While these early attempts have shown promising potentials for data mining applications, there are still limitations to overcome, such as high ANN training noise and low hotspot detection accuracy, etc.

To better address the issues above, we introduce the concept of critical hotspot feature as an effective representation of the original pixel based design layouts. Compared with 2D pixel images used in [83, 92], critical features reduce the dimension of the original data samples and filters out detection noise. With powerful Machine Learning Models, our flow demonstrates high speed with good detection accuracy and small false alarm rate (10% of actual hotspots).

The key contributions of Section 2.2 are summarized as follows,

• We propose a CAD flow with effective *critical feature* extractions for machine learning noise reduction, accurate hotspot detection and fast speed.

• We employ supervised training algorithms to establish machine learning models with special techniques for further accuracy improvement.

• We achieve good scalability to new RETs and new manufacturing conditions, ensured by fast/flexible retraining of the learning models.

In the following subsections, we will first introduce some preliminaries then describe each step of our proposed detection flow in detail. Performance enhancement techniques will also be developed and assessed. Testing bench-

marks and simulation results will be presented and discussed followed by a brief summary at the end of this section. Preliminary results of this work were presented at the *IEEE International Conference on IC Design Technology (ICICDT)* 2009.

2.2.1 A Motivational Example

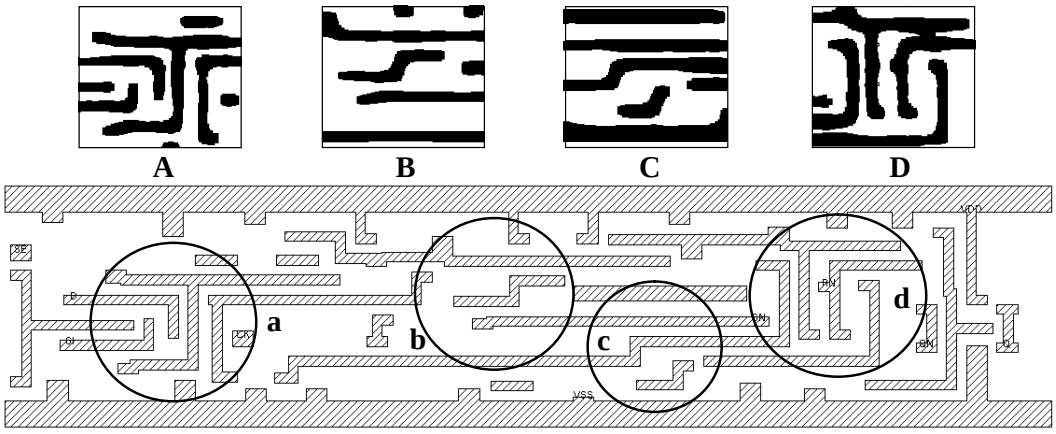


Figure 2.5: a,b,c and d are samples from a $45nm$ design [2]; A,B,C and D are their respective litho-simulated images after OPC.

The rationale of our proposed hotspot detection flow can be illustrated from Fig. 2.5, where a, b, c, and d are metal1 layout patterns for a certain $45nm$ logic cell design; correspondingly, A, B, C, and D are the litho-simulated print-image of a, b, c, and d after OPC, representing post fabrication layouts of the 4 local patterns in the original design. From A, B, C and D, CALIBRE [1] shows that a and d are highly susceptible design patterns to the fabrication process, pattern c is slightly better and pattern b contributes the least towards

generating hotspots in post fabrication stage. With A-D, we can re-design a, c and d area in the original layout to avoid post fabrication circuit defects, such as shorts, opens or other issues that high variability brings. However, in order to get A-D, there are complex integrals and convolutions involved in lithographic simulation, which is very expensive in terms of both run-time and computational resources, especially when invoked repeatedly for DFM closure.

In this subsection, a machine learning kernel is established through mining the correlations between a small set (a few hundred) of design layouts such as a to d and their corresponding lithographic simulated images as in A to D. Afterwards, the kernel is used to predict post fabrication hotspot patterns without invoking lithographic simulations, leading to very fast detection speed. Unlike some previous work [83, 92] which focus on pixel based layouts, we propose a set of *critical features* as a compact representation of the original images. Algorithms for extracting these features are fast and hotspot capture rate is proven high (plus 90% on average) by a large set of simulation benchmarks.

2.2.2 Machine Learning based Detection Flow

The overall flow is shown in Fig. 2.6, consisting of 6 stages.

In *Stage I*, a small set Ω of design layouts are established as initial raw data set of *MLK* training, followed by *Stage II*, where a collection set Δ of typical binary pattern images are sampled from the design layout set Ω. In *Stage III*, CALIBRE simulation set-ups (such as technology process, design rules,

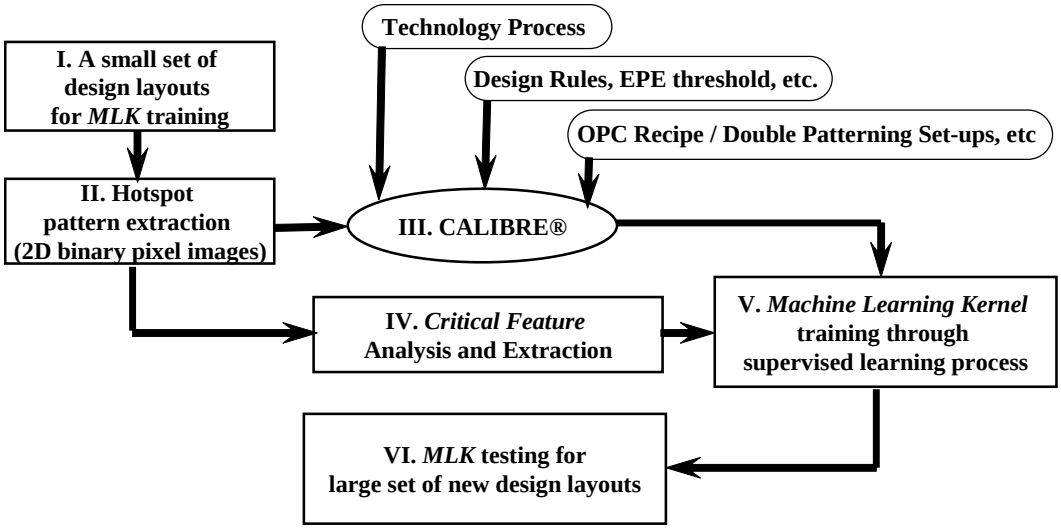


Figure 2.6: Overall proposed flow with stages *I-VI*

OPC, EPE, etc) are loaded and lithographic simulations are performed on set Δ at a one-time cost, identified post-OPC hotspot patterns are stored in set Θ according to EPE threshold. *Stage IV* performs *critical feature* analysis and generates the critical metrics vector, as training and classification input vector to the *MLK*, which is an essential step for low noise data-training/classification and high hotspot detection accuracy. *Stage V* imports the critical metrics vector from *Stage IV* and performs supervised *MLK* training and validation based on set Θ from *Stage III*, resulting in an optimized highly non-linear *MLK*. The established *MLK* is tested on large sets of new design layouts in *Stage VI* with the same setups as in *Stage III*.

2.2.3 Critical Feature Analysis and Extraction

We define the *critical hotspot feature* as a metric extracted from the original design layout pattern to represent the set of parameters most critical and sensitive to the occurrence of lithography hotspots. An effective *critical feature* should remain the same under arbitrary 2D transformations such as shifting, rotation and mirroring, etc. A generalized set of *critical feature* includes all pixel based image analysis transforms and representations, such as discrete fourier transform, Hough transform, distance transform and representations of particular patterns of interest.

We propose 3 features, namely *Bounded Rectangle*, *T-shape metal* and *L-shape metal* features. As a compact representation of the original pixel image pattern, these features capture the relative geometry relations in between metal tracks effectively and lead to satisfactory detection accuracy. Unlike other 2D pixel based transforms, the proposed features are computationally easy to extract, thus contribute to a fast detection flow. Although more features can be added to enrich the existing critical feature metric, they can also slow down the detection process. Thorough simulations demonstrate that the proposed 3 critical features are generic enough to cover up to over 90% of hotspots with small false alarms (less than 10%).

Unlike previous work [83, 92] that directly employ binary layout pattern images, *critical feature* enables us to zoom closer onto the key parameters for hotspot detection thus greatly reduces the time, complexity and classification noise in the supervised training.

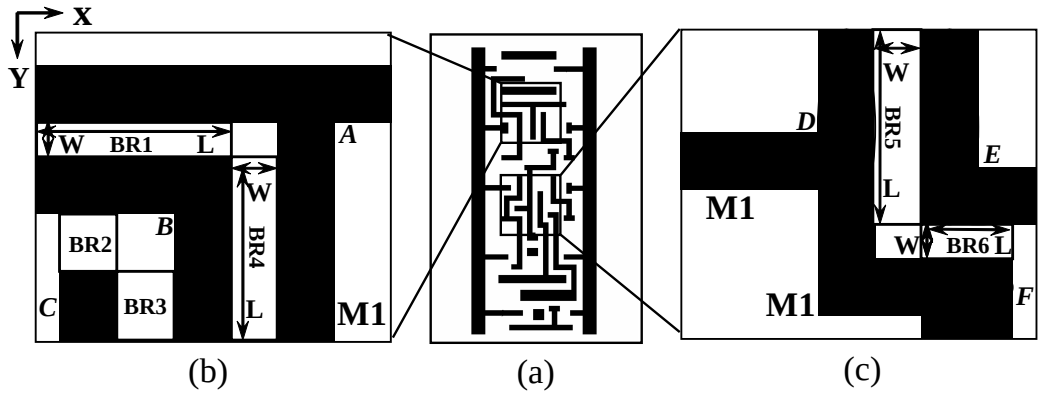


Figure 2.7: (a) A 45nm design layout; (b)(c) Two example sample patterns under the signature extraction process.

2.2.3.1 Bounded Rectangle Feature

The first *critical feature* we propose is the *Bounded Rectangle* feature, as illustrated by the rectangles in between of metal wires in Fig. 2.7(b)(c), *BR1* to *BR6*. Fig. 2.7(a) is a certain design layout, with two sampled patterns (b)(c), from which critical features are to be extracted. *BR* feature records the relative geometrical positioning between adjacent metal tracks through metal interval representation. Each *BR* is expressed with a 5 parameter vector *(W, L, X, Y, D)*, where L denotes the length of *BR* along the metal edges confining itself; W denotes the width of *BR* along the direction perpendicular to L; *(X, Y)* is the coordinates of the upper-left corner of *BR*; D is set to 0 if W is along X direction, to 1 if W is along Y direction. For example in Fig. 2.7(b), *BR1.D=BR2.D=1; BR3.D=BR4.D=0.*, in Fig. 2.7(c), *BR5.D=0; BR6.D=1.* Detailed *BR* extraction algorithm is shown in Algorithm 1. Here, patterns in set Δ are formatted to N by N binary pixel images.

Algorithm 1 *Bounded Rectangle Feature* Extraction

```
1: //Rectangle Scanning and Harsh-Table Build-up
2: //input pattern is N pixel by N pixel image from set Δ
3: for y from 1 to N do
4:    scan intervals divided by metal tracks on row y
5:    store (x,width) pair for each interval
6:    for each (x,width) pair do
7:       update hash table T; x as key,(y,width) as value
8:    end for
9: end for
10: //BR feature extraction
11: for  each x in key list of T do
12:    for  each (y,width) pair in T[x] do
13:       if width(y) == width(y++) then
14:          update sequence S with current record
15:       else if width(y) != width(y++) then
16:          store width in W, sequence length of S in L
17:          store current x in X, y in Y; set value for D
18:          reset S
19:       end if
20:    end for
21: end for
22: return list (W,L,X,Y,D)
```

2.2.3.2 T-shape and L-shape Features

The other two *critical features* we propose are the *T-shape* metal feature and *L-shape* metal feature, as illustrated in Fig. 2.7(b)(c), region A to F. *T-shape* and *L-shape* features signify the number of *T-shape* and *L-shape* metal wires extending along both sides of *BRs*, together with corresponding wire width and jog width information. For example in Fig. 2.7(b), area A is *T-shaped* metal for *BR1/BR4*, area B is *L-shaped* metal for *BR1/BR2/BR3/BR4*, area C is neither *T-shape* nor *L-shape* for *BR2/BR3*. In Fig. 2.7(c), area D

Algorithm 2 *TL-shape* Feature Extraction

```
for each BR i do
  if D == 0 then
    scan metal track widths on both sides of i along X direction, with range [Y,Y+L]
  else if D == 1 then
    scan metal track widths on both sides of i along Y direction, with range [X,X+L]
  end if
  i.T = scale_mapping(T-shape ambience metal tracks)
  i.L = scale_mapping(L-shape ambience metal tracks)
end for
return list (T,L)
```

is *T-shape* metal track for *BR5*, area E and F are *L-shape* metal tracks for the right side of *BR5* and both sides of *BR6*. Thus after the *TL-shape* feature extractions algorithms are performed, we have the following: *BR5.Tf=1, BR5.Lf=1; BR6.Tf=0, BR6.Lf=2*. More fine-grind quantization method could be employed to differentiate polygon width and jog width for the *TL-shape* feature. Details are shown in Algorithm 2.

Combining these features, we obtain the critical information of both adjacent metal tracks and the intervals in-between them for a certain pixel based layout. These features proposed are proven to have high detection efficiency meanwhile maintaining a low noise figure compared with [92]. Through executing proposed *critical feature* extraction, a feature metric vector is derived for each bounded metal track region in every image from set Δ, in the form of *(W, L, X, Y, D, Tf, Lf)*. A sorted collection of these vectors forms a metric for each layout image sample, such metric is the final input for MLK

supervised training in the following *Stage V*. The purpose of the training is to iteratively establish a knowledge kernel, which can be applied later to predict hotspot patterns for new design layouts.

In order to reduce the dimensionality of the extracted *critical feature* matrix and eliminate potential interrelated variables meanwhile retaining as much as possible of the variation present in the matrix, Principal Component Analysis is performed and principal component *critical feature* matrix is derived for more effective *MLK* training and classification.

2.2.4 Machine Supervised Training

As an important classification technique in data mining, Artificial Neural Network (ANN) originated from imitating human brain neuron networks and human learning activities [118]. A small size ANN with 5 to 10 hidden layer neurons can be trained to establish highly nonlinear models with the highly interconnected manner of neurons in the network. Since the machine learning algorithms and functions here are directly invoked from environment [4], we omit the corresponding details. In later sections, we will further explore and fine-tune the algorithmic details of various learning models.

In our flow, we take the extracted critical features as inputs and feed into the ANN network for supervised training, which is essentially parameter optimizations for all the neurons in the network, through iterative coefficient updating. After the training procedure, in Subsection 2.2.6 the established ANN kernel will be directly applied onto various benchmarks in $45nm$ tech-

nology for hotsot detection tasks.

As an illustration to ANN working mechanism, Fig. 2.8 shows a system block of ANN on the left, with a single neuron on the right.

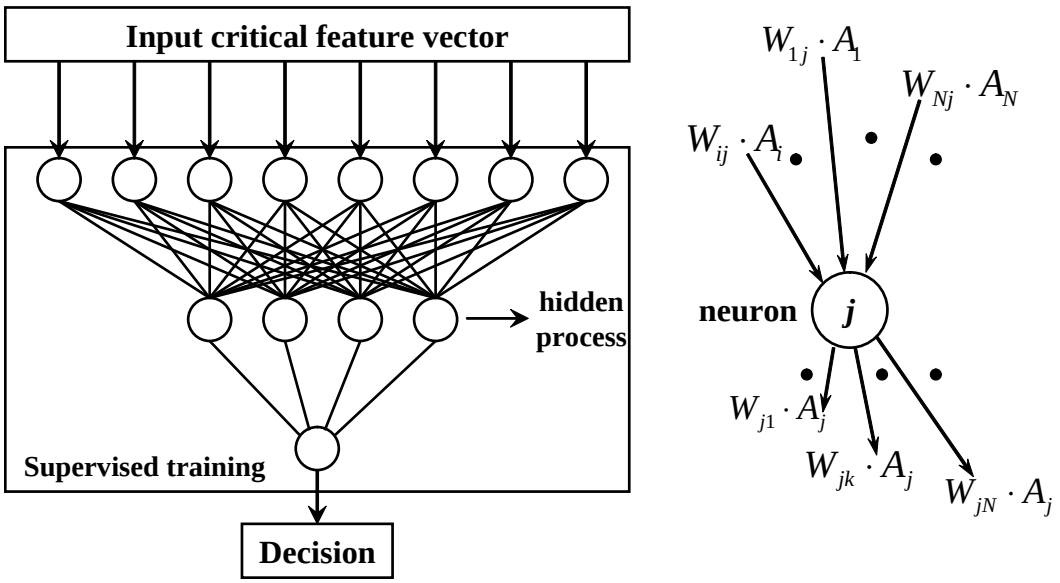


Figure 2.8: Typical structure of the ANN network and an individual neuron

As shown in Fig. 2.8, supervised training for ANN results in a weight vector on all the links of the neuron network to minimize the MSE (mean square error) between network prediction results (binary decision) and the given supervising target (accurate lithography simulation result for the same layout from which critical features are extracted). As an iterative process, it employs various kernel functions for weight update towards MSE minimization. In Fig. 2.8, W_{ij} is the weight on link i for the *jth* neuron, and neuron j's final

output value A_j is fed to all next level neurons, which is defined as follows,

$$A_j = f(\sum_{i=1}^{N}(W_{ij}A_i) + \theta_j)$$

where θ_j is defined as bias, $f(\ )$ is the judgment function, such as a *sigmoid* function or step function, etc.

The objective of the iterative training process is simply written as:

$$Minimize \qquad \frac{1}{M}\sum_{p=1}^{M}(A^p_{output} - T^p_{calibre})^2$$

where M is the total number of samples in the supervised training set, A_p is the ANN prediction output for *pth* sample pattern at certain iteration and T_p is the training target of the sample, which is pre-set by lithographic simulations. For each iteration, all training samples are employed and prediction is made based on the most updated set of weights. Minimization stoping criteria is the satisfaction of certain pre-defined training error target. For effective weight update, various network architectures and update functions have been proposed, a general form of the update is as follows,

$$x_{k+1} = x_k - \alpha_k g_k$$

where x_k is the weight vector for the network links and x_{k+1} is the updated vector after 1 iteration and α_k is the learning rate vector for x_k. g_k is the update function which we can choose from a wide range of packages in MATLAB, such as resilient backpropagation and conjugal gradient, etc.

The supervised training in for our proposed detection flow is set as in Fig. 2.9, where ANN inputs the *critical feature* vector and outputs the

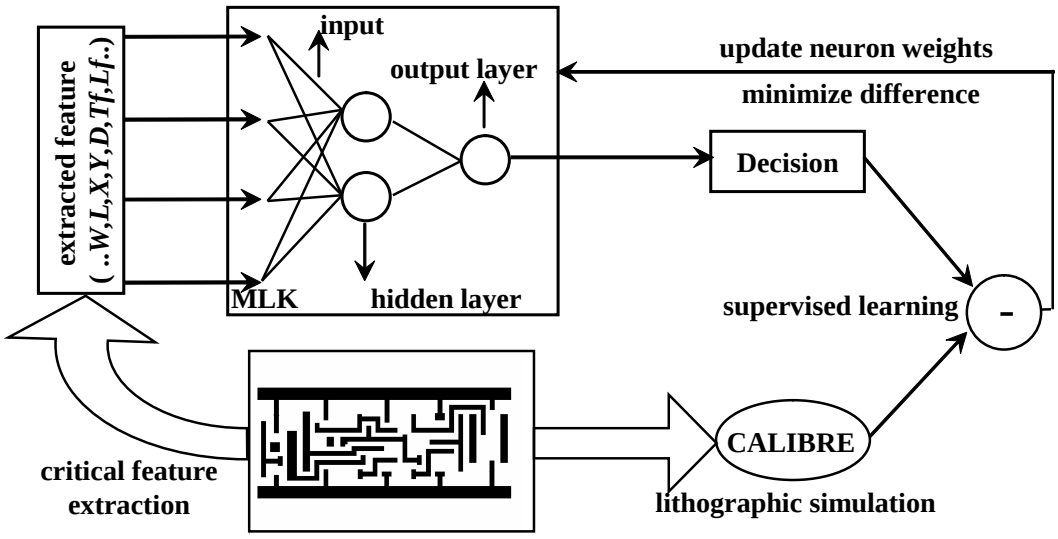


Figure 2.9: An illustration of the machine supervised training process

final detection decision (1 for hotspot, 0 for non-hotspot); network neuron coefficients are then iteratively updated under the supervision of CALIBRE simulation results towards MSE minimization. During the training process, various parameters need to be fine-tuned, such as the sampling window size, hidden layer neuron number, learning rate, number of epoches, training error target, kernel function and gradient update algorithms.

2.2.5 Detection Enhancement for Complex Design Layouts

For large area design layouts, we propose an algorithm to further increase detection accuracy, which we refer to as proximity detection. *detection_effort* in Algorithm 3 is defined as follows,

$$e = \frac{\sum_{effective\_area}^{i} SPT(i)}{whole\_design\_area}$$

Algorithm 3 PDA (Proximity Detection Algorithm)

```
1: //Generate locations for Sampled-Pattern-for-Testings
2: Generate SPT seed (detection_effort, distribution)
3: Populate each seed locally (proximity_effort)
4: //Invoke ANN kernel with multi-threaded technique
5: for each SPT(i) do
6:    MLK(SPT(i).feature)
7: end for
8: hotspot voting process within each local proximity
```

where e is the ratio of total effective areas covered by SPTs (sampled patterns for testings) in Algorithm 3 versus original design layout area. Fig. 2.10 shows an illustration of Algorithm 3, where we examine a $45nm$ cell design layout. With Algorithm 3, false detection on $SPT3$ and $SPT8$ are detected and discarded through a voting process among their respective proximity SPTs before final hotspot decisions are made for the particular region. Advantages of combining our proposed flow with Algorithm 3 is that it further improves detection accuracy using proximate pattern correlations. Since our ANN (artificial neural network) kernel is already established, extra prediction time caused by the SPT area overhead is very little, especially with multi-threaded techniques.

2.2.6 Simulation and Testing

Our testing benchmarks are from a $45nm$ technology [2], with simulations performed on 2.4GHz Intel Xeon Linux with 4GB memory.

Altogether, 70 CALIBRE identified hotspot patterns and 64 non-hotspot

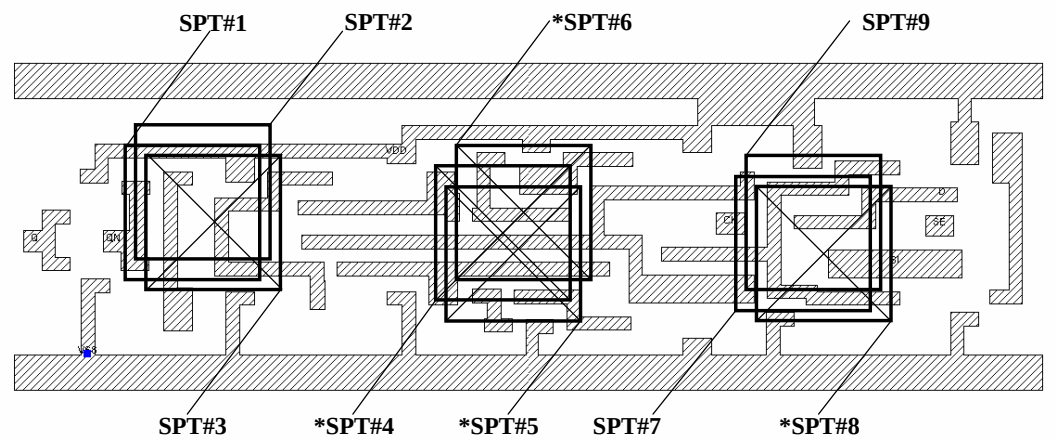


Figure 2.10: An example of cell level hotspot detection using Algorithm 3. (e set to < 0.1 for observation convenience)

patterns participated in the ANN supervised training, with window size less than 1.5 micron by 1.5 micron. ANN hidden layer neuron is set to 5, learning rate to 0.05, epochs to 30, training error target to 0.01, kernel function is finalized to Resilient Backward Propagation with sigmoid function. The entire training process took less than 200 seconds to finish.

Upon completion of the supervised training, we apply the ANN kernel to the following benchmarks for cross-validation and testing:

- B1: Small Area Patterns. B1 consists of 35 small area design patterns from [2], which the ANN kernel has never seen during training and validation. As a comparison baseline, CALIBRE litho-simulation was carried out to label the real hotspot patterns and non-hotspot patterns.

- B2: $45nm$ Cell Level Layout Patterns. B2 consists of 5 large area cell

level design layouts from [2], which did not participate in the supervised training step of the ANN kernel.

- B3: Chip Level Design Layout. B3 is taken from a $45nm$ open-source FFT IP core that is fully placed and routed with an industry APR tool. It has a total of 644 modules and 48027 standard cell instances placed on a 380 by 350 micron die.

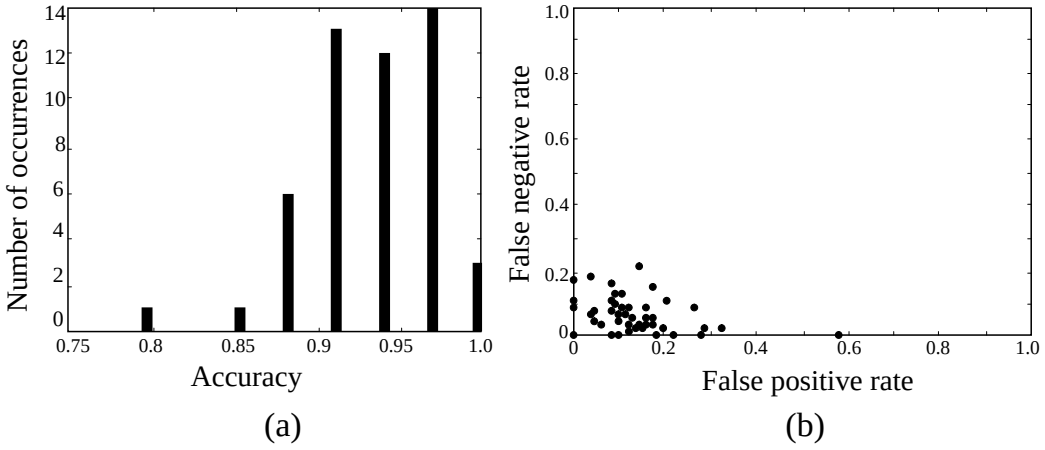


Figure 2.11: (a) Histogram of B1 hotspot detection accuracies with 50 independent experiments; (b) *False positive rate* versus *false negative rate*

From Fig. 2.11(a)(b), we fully evaluate the established ANN kernel with 50 independent cross-validations on B1. As a result, our flow demonstrates above 90% of accuracy with over 85% confidence, with the best detection accuracy 100% and worst 80% (only one occurrence every 50 testings). In Fig. 2.11(b), we plotted another two crucial parameters for performance evaluation, namely *false positive rate* and *false negative rate*. *False positive rate*

is the percentage of actual non-hotspots detected as hotspots (hotspot false alarms); *False negative rate* refers to the percentage of actual hotspots detected as non-hotspots, which is to be kept minimal.

For B1, our average *false positive rate* is 15%, and *false negative rate* is 6%, as shown in Fig. 2.11(b). Detailed summary is in Table 2.1. Since B2/B3 are large area design layouts, we employ the proposed proximity detection algorithm for accuracy enhancement. e is set to <1.0 for B2 and between 1 to 3 for B3. e larger than 3 is considered as high effort, meaning the total SPTs area in Algorithm 3 is more than 3 times of original layout area. e should be set higher for denser design layouts.

As a result, hotspot detection accuracy for B2 reached 100%, with *false positive rate* 0%. B3 is evaluated with the same manner and results show plus 90% of hotspot detection accuracy with very small (10% of total hotspots) *false positive rate* (detection false alarms). More experimental result details can be found in Table 2.1.

2.2.7 Summary

As crucial assistance for accurate lithographic simulation, fast and high fidelity machine learning based hotspot detection starts playing a more important role in modern DFM CAD due to many of its desirable characteristics. In this section, we have presented a *critical feature* extraction/classification flow for the detection of lithography hotspots utilizing powerful machine learning techniques. With the *critical feature* representation, learning/detection noise

for the training procedure is effectively reduced without run-time overhead when compared with previous works [92]. Experimental results demonstrate small detection false alarm rate (10% of actual hotspots) and an average of plus 90% detection accuracy on the whole-chip scale testing benchmark. For certain smaller testing cases, our proposed detection flow shows good accuracies with best case 100%. Total run-time for the proposed flow is around 5 minutes for the largest testing benchmark on a 2.4GHz dual-core Linux workstation.

Table 2.1: Performance of the proposed hotspot detection flow

Benchmarks	Testing target	HotSpot Detection Rate	Non-HotSpot Detection Rate	HotSpot Detection -False-positive Rate	HotSpot Detection -False-negative Rate
B1 [a] (MLK kernel; without PDA[b])	Best Perf	100%	98%	2%	0%
	Average Perf	94%	85%	15%	6%
	Worst Perf	80%	54%	56%	20%
B2 [a] (using PDA[b])	XNOR2_X2	100%	100%	0%	0%
	DFF_X2	100%	100%	0%	0%
	SDFFRS_X2	100%	100%	0%	0%
	SDFF_X2	100%	100%	0%	0%
B3 [a](with PDA[b]):	Accuracy Rate	90%	—	10% false alarms	10%

[a] Hotspot detection EPE threshold is set to $15nm$ in $45nm$ technology process.

[b] PDA: Proximity Detection Algorithm

2.3 High Performance Hotspot Prediction With Successively Refined Machine Learning

In the process to bridge the wide gap between design demands and manufacturing limitations of the current mainstream 193nm lithography, various Design for Manufacturability (DFM) techniques [27, 34, 60, 89, 97] have been proposed to improve product yield and avoid potentially problematic patterns (i.e., process hotspots). However, for $45nm$ node and below, hotspot patterns still exist even after Design Rule Checking (DRC) and various advanced resolution enhancement techniques (RET) such as Optical Proximity Correction (OPC), Double Exposure Double Patterning Lithography (DEDPL)[16] and Self-Aligned Double Patterning Lithography (SADP)[10, 80]. While these RETs greatly improve the IC manufacturing yield, the remaining hotspot patterns become much more difficult to detect at design stage and costly to fix at sign-off verification stage.

Consequently, faster and higher performance hotspot detection engines can play an essential role to further enhance physical verification/DRC, and to develop smart (lithography-friendly) physical design tools. As we have motivated in the previous section, on the one hand, conventional approaches that employ lithographic simulations [67, 103] are accurate but very costly to run; on the other hand, approaches that utilize pattern/graph matching techniques [66, 135, 138] are fast but reliant on a set of pre-defined hotspot patterns. However, general hotspot patterns are hard to define/model in a deterministic manner. Too many patterns lead to high over-estimate (false alarms) and

too few patterns result in low hotspot coverage. Pattern enumeration could become more problematic as process technology advances and RETs improve, as the definition of the real lithography hotspots is highly dependent on the evolving manufacturing conditions.

In recent years, there are emerging works starting to incorporate modern data mining methods for fast and accurate hotspot detection. A neural network judgment based detection flow was proposed in [92], where 2D hotspot image patterns were directly used to train an artificial neural network (ANN) model. In [83], data mining algorithms are developed for hotspot pattern (2D images) clustering. While these early attempts have shown promising potential for lithography hotspot detection using data mining methods, there are still limitations to overcome, such as high noise and low accuracies.

Later in [47], a support vector machine (SVM) based hotspot detection method is utilized through performing 2D distance transform and histogram extraction on pixel based layout images. Also in [132, 133], SVM is employed for hotspot detection through extraction and classification of certain special layout density related metrics. [47, 132, 133], as improvements over [92] and [83], demonstrate higher detection accuracy and lower classification noise, due to the introduction of high fidelity metrics. However, these approaches have limited efficiency in run-time and/or detection coverage, since 2D transforms and density extractions can be quite expensive to perform, meanwhile detection windows (or hotspot candidate locations) for the layout images can be very hard to anchor for full-chip area detections. In practice, these windows

are slid, scanned or sampled across the entire layout with a certain amount of overlap (or blank interval) between each other. As an inevitable result, detection performance becomes a trade-off between run-time and detection coverage. In the preliminary work [43] of Section 2.2, we proposed the *critical hotspot signature* that are extracted through certain special edge-based metrics from IC physical layouts. Although such edge-based extractions operate much faster compared with [47, 133], their chip level applications still face similar problems such as scanning window coverage, etc.

Moreover, very few existing studies deal with the detection challenges under the real manufacturing conditions in which hotspots become increasingly harder to detect. In order to be practically employed in modern IC physical design, a successful hotspot detection engine must demonstrate superior speed compared to full lithography simulation ($>$ 100 CPUs running in the order of days) and DRC (tens of CPUs running for a few hours), as well as comparable performance to meet the real design and manufacturing requirements. Unfortunately, under such situations, the hotspot evaluation models in [43, 47, 83, 92, 133] suffer from severe performance degradation. This is because detecting real hotspots under industry-strength PDK/manufacturing conditions requires more than just one straightforward model, but multiple levels of identification models for performance refinement.

To better address the issues and challenges above, we extend our work in [42] and propose a generic hotpot detection methodology that is capable of fast on-line data learning and high performance hotspot pattern identifications.

This methodology provides a full layout, feature-centric analysis without being penalized in run-time or coverage by conventional sliding window [133] or raster scanning [47] related techniques. Under such a framework, we define novel layout analyzing algorithms that process the layouts in a fragment-based manner. We implement the framework in a leading industrial geometry processing engine via a shared object library [1]. The proposed framework is tested with enhanced ANN and SVM on large industry layouts under real manufacturing conditions, demonstrating very promising performance in detection accuracy, false-alarm suppression and CPU run-time.

The rest of the section is organized as follows. In Subsection 2.3.1, we further motivate a few key challenges in lithography hotspot identification and summarize our major contributions. Subsection 2.3.2 gives an overview of our proposed methodology. In Subsection 2.3.3, we propose a *Layout Analyzer* for the extraction of hotspot related features with high detection coverage and speed, followed by Subsection 2.3.4, where we describe in detail our *Hotspot Identifiers* with special machine learning models for enhanced accuracy. In Subsection 2.3.5 we propose a novel flow to integrate, configure and validate multiple successive levels of hotspot identifiers for ultra-low false alarms under current real manufacturing conditions. Simulation results on various placed and routed industry layouts are assessed and analyzed in Subsection 2.3.6. Subsection 2.3.7 provides a brief summary of this section.

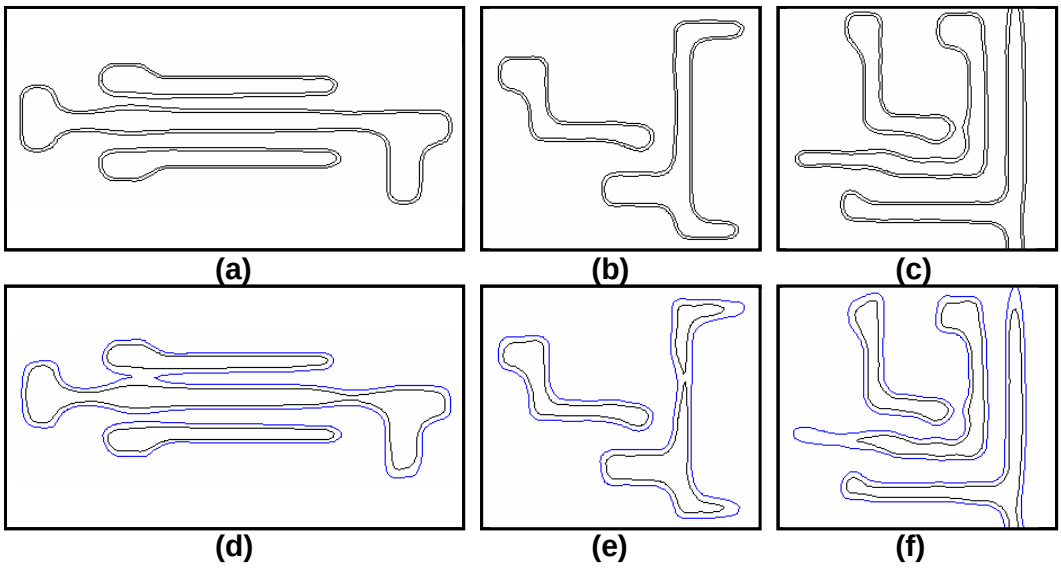


Figure 2.12: An example of process variability bands depicting manufacturing variations under different manufacturing conditions [119]

2.3.1 Motivation and New Contributions

To visualize the aforementioned challenges, in Fig. 2.12 we show the printed images of three layout patterns under two different manufacturing conditions: (a)-(c) are printed under a real production $45nm$ process which provides insights on the frequency of occurrence of hotspots under real manufacturing conditions, while (d)-(f) are under simplified but widely accessible manufacturing conditions (e.g., freePDK45nm) which lack the detailed information in optical, resist and actual OPC and RET recipes used during production tending to exacerbate process variations. In this case, we observe significantly less poor-printability areas in (a)-(c) than in (d)-(f).

The significance of this motivational example is manifold: first, a real

hotspot pattern by definition is strongly dependent on manufacturing conditions; second, the number of real hotspots under continuously improving manufacturing conditions becomes less (assuming improving RETs) but hard to fix at post-layout stages. Moreover, non-hotspot detection inaccuracy (false-alarms) increases the burden of hotspot correction processes, and should therefore be minimized. Last but not least, the huge data volume of large area design layouts requires ultra-fast detection speed and good run-time scalability.

Here we summarize the key challenges for detecting lithography hotspots under the real manufacturing conditions as follows:

- Hotspots are highly dependent on the real manufacturing conditions.
- Hotspots become harder to identify in the early design stages and harder to fix at the post-layout correction stages.
- Non-hotspot detection accuracy (false-alarms) becomes important as excessive false-alarms are severely penalized in post-layout corrections.
- Ultra-fast detection speed is desired for large layouts and for guiding lithography-friendly physical design.
- Sliding window techniques can be highly penalized by considerable loss of accuracy and detection coverage.

To address these challenges, we propose a novel methodology and several highly effective techniques as follows:

- We propose a generic methodology for lithography hotspot detection that is compatible to evolving RETs and manufacturing conditions.
- We define special hotspot signature measurements for ultra-fast, full layout detection without sliding window or raster scanning techniques.
- We introduce fast layout analyzers and generic hotspot pattern identifiers with powerful machine learning models for high fidelity detection.
- We develop a generic and efficient flow with multiple levels of successive pattern identifiers for ultra-low identification false alarms.
- We perform thorough qualification using real industry examples for a $45nm$ METAL1 process under real manufacturing conditions.

2.3.2 Methodology Overview

First we define several important terms used throughout this dissertation, as in equations Eqn.(2.2) to (2.5):

Definition 2.3.1. ***Hhit***: the hotspot detection accuracy rate.

$$Hhit = \frac{correctedly\_detected\_hotspots}{real\_hotspots} \tag{2.2}$$

Definition 2.3.2. ***Hmiss***: the hotspot detection inaccuracy rate.

$$Hmiss = \frac{undetected\_hotspots}{real\_hotspots} = 1 - Hhit \tag{2.3}$$

Definition 2.3.3. ***Hextra***: the hotspot detection overshoot (false-alarm rate).

$$Hextra = \frac{falsely\_detected\_hotspots}{real\_hotspots} \tag{2.4}$$

Definition 2.3.4. ***Nhit***: non-hotspot detection accuracy rate.

$$Nhit = \frac{correctedly\_detected\_nonhotspots}{real\_nonhotspots} \tag{2.5}$$

Before going to further details, we first illustrate an overview of our proposed methodology in Fig. 2.13, which is divided into the *calibration stage* and the *detection stage*.

The *calibration stage* involves: (1) a relatively small set of layouts for configuring the *Hotspot Identifers* via supervised learning techniques; (2) a *Layout Analyzer* for characterizing layout geometries; (3) a lithography simulator (or post-silicon measurement) to provide accurate information of the real hotspots as learning targets; (4) a novel calibration process for the training and validation of *Hotspot Identifier* models via successive refinements. The result of the calibration stage is a set of *Hotspot Identifiers* established at a one time computation cost. In the *detection stage*, we apply the *Hotspot Identifers* to search for hotspot patterns over very large volume of design layouts with high efficiency given that the identifiers have been setup *a priori*. This stage mainly consists of: (1) a *Layout Analyzer*; (2) a hotspot detection process that utilizes the multiple *Hotspot Identifiers* via levels of refinements.

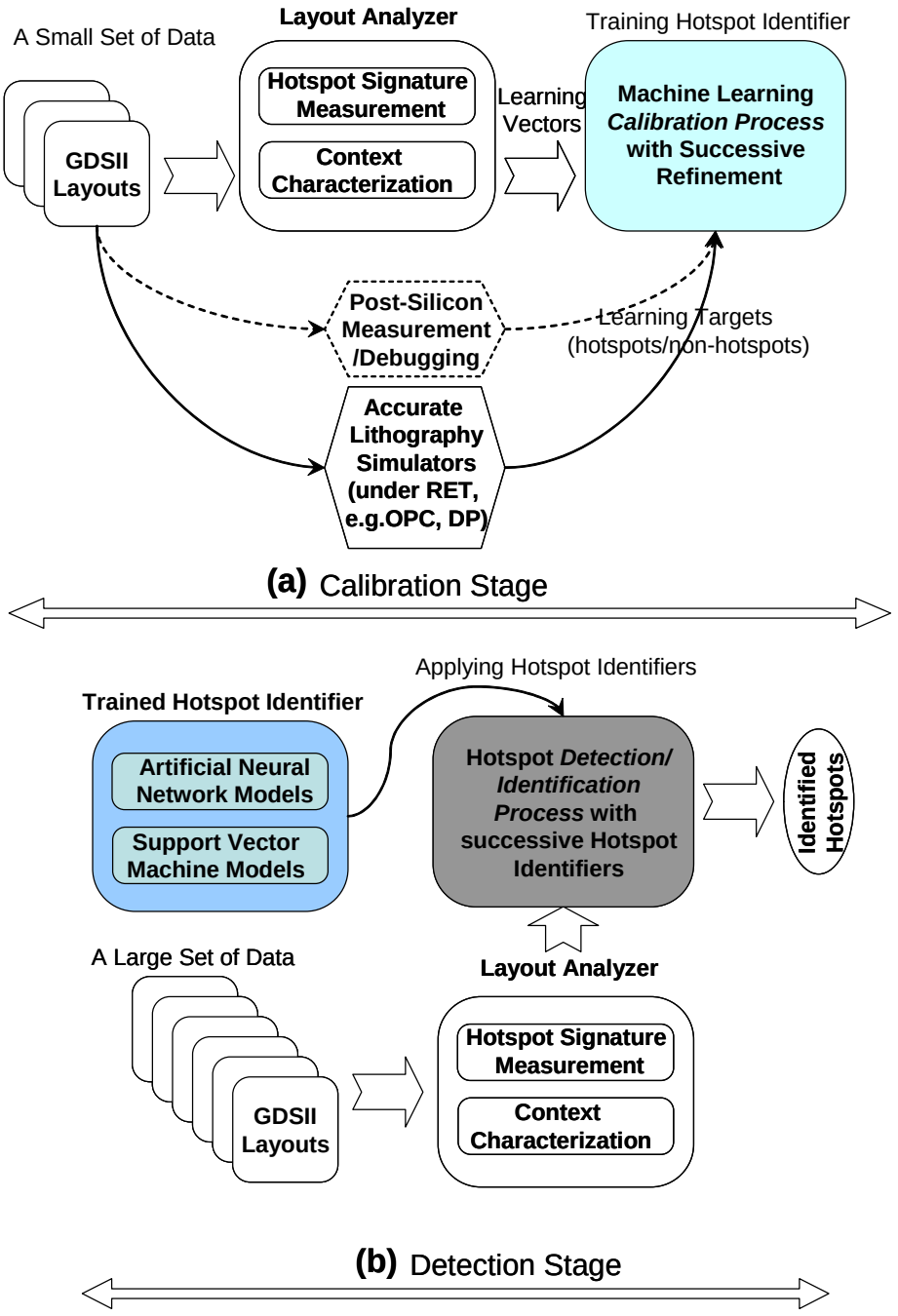


Figure 2.13: Our proposed hotspot detection methodology consisting of: (a) the *Calibration Stage* and (b) the *Detection Stage*

Note that the function of *Layout Analyzer* is to characterize the layout and convert patterns to compact 1D vectors. With these data vectors, special *Hotspot Identifiers* can be : (1) trained and validated under the supervision of our employed production simulator [1]; (2) applied with very high speed and fidelity over new layouts. Note that the *Hotspot Identifiers* must be updated by re-running the *calibration stage* if the design rules or manufacturing conditions are changed for the new layouts. In the following subsections, we will further explain the details of each block from Fig. 2.13.

2.3.3 Feature-Centric Layout Analyzer

Layout Analyzer performs the function to characterize the layout context and extract hotspot related features in the format of data vectors. Here we define *hotspot feature metrics* (or *hotspot signature measurements*) as a set of special measurements which contribute strongly to the decision making process of hotspot detection. Unlike special restricted design rules, a *Layout Analyzer* does not decide whether a certain pattern is hotspot or not, but it leaves the decision making process to the recursively refined supervised training of *Hotspot Identifiers* using machine learning techniques. In face of the aforementioned challenges under real manufacturing conditions, a successful *Layout Analyzer* must first define a proper set of *hotspot signature measurements*. Unlike previous studies utilizing 2D transforms or density calculations or sliding window techniques, we propose novel metrics and special data structures for significant run-time reduction and satisfactory accuracy.

2.3.3.1 Hotspot Signature Measurements

With our special *hotspot signature measurements*, we also define several types of layout measurement operators, which are designed with good scalability to cover the entire layout without applying sliding window or density based techniques. We illustrate the different types of measurements in Fig. 2.14, namely (a) corner information (convex or concave), (b) distance to an externally facing polygon edge and (c) distance to an internally facing polygon edge. In a fragmented design layout where the fragments (polygon edges) are indexed numerically, these measurements can be programmed and optimized to reach very high speed and memory efficiency using the Advanced Programming Interface of [1]. With a proper combination of these measurements, we can accurately characterize the entire layout at a one-time cost. Therefore, the context representation of each layout geometry is transformed to a table lookup problem that can be solved in constant time. With the run-time advantage, we are able to analyze every geometry (jogs, corners, intra/inter-distances, etc.) in the design layouts with full coverage, which is otherwise difficult to achieve using sliding window or density-based methods.

Table 2.2 details the different types of measurement operators. In par-

Table 2.2: Hotspot signature measurement operators

operators	operation description (features to measure)
$f_{corn}(\cdot)$	corner information: CV(convex) / CC(concave)
$f_{ext}(\cdot)$	external inter fragment distances
$f_{int}(\cdot)$	internal inter fragment distances
$f_{misc}(\cdot)$	miscellaneous information

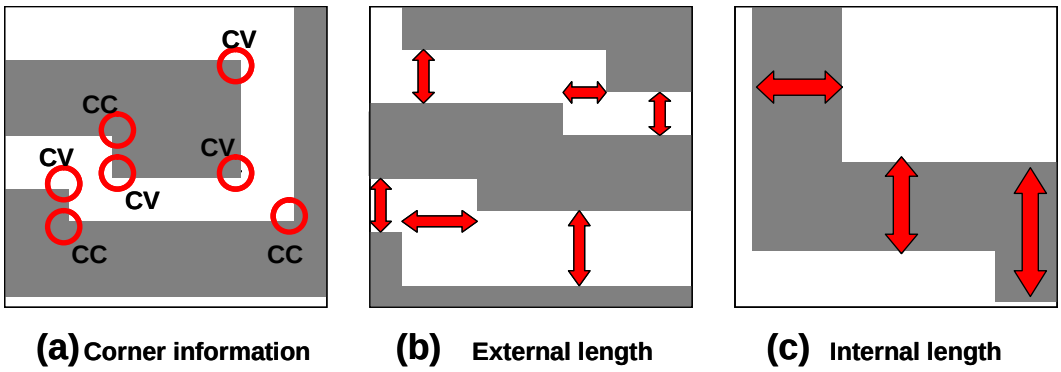


Figure 2.14: 3 major types of hotspot feature measurements

ticular, *Operator* $f_{misc}(\cdot)$ requests extra information regarding $Frag$, such as fragment orientation (X or Y axis) and the length of $Frag$, etc. Consequently, hotspot related geometries of a design layout are represented and indexed in high resolution (per-fragment based) with a combination of the defined operators, leading to a high fidelity quantization process to be detailed as follows.

2.3.3.2 Fragmentation based Context Characterization

The ultimate goal of *Layout Analyzer* is to provide high resolution layout characterization with full scanning coverage. To achieve this goal, our *hotspot signature measurements* are carried out on a per fragment basis. Given a properly fragmented layout and any fragment of interest F, we illustrate the concept of an effective radius r in our proposed context characterization procedure. As shown in Fig. 2.15(a)(b), r centers at (each) fragment F. By definition, effective radius r covers the neighboring fragments which need to be considered in the context characterization of F. In its empirical nature, r

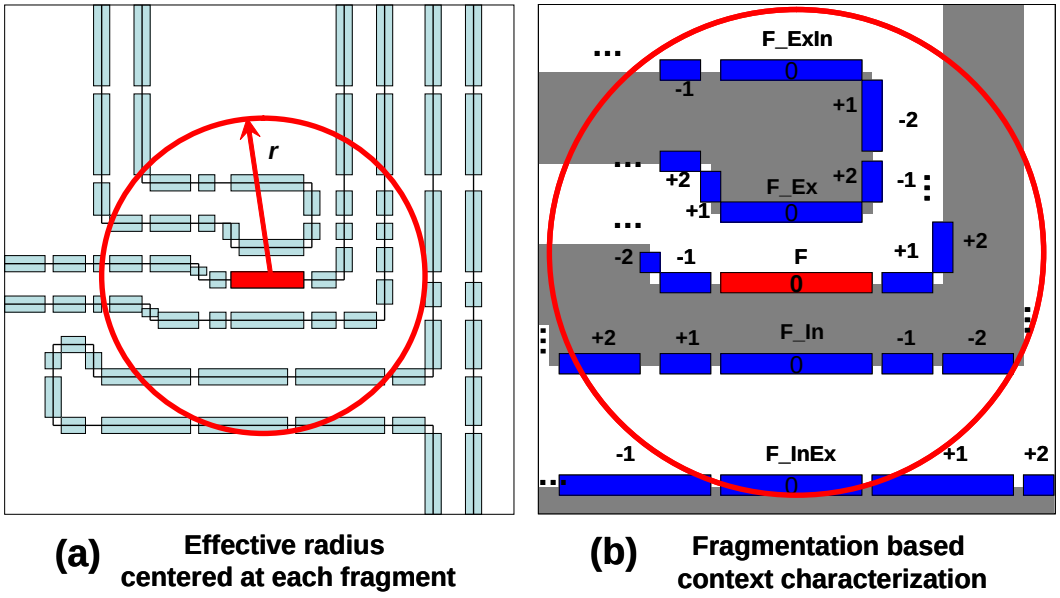


Figure 2.15: Fragmentation based hotspot signature extraction

depends on the lithography processes, and it is generally easy to pick in the training and validation procedure given the manufacturing conditions. With Fig. 2.15(b), we illustrate the context characterization step as follows.

First, suppose the current fragment of interest is F (colored red in the center of the effective circle), we define several short-hand notations whose

Table 2.3: Short-hand notations for fragmentation

notation	descriptions of the short-hand notation
F	current fragment of interest (detection anchor point)
F_In	the fragment(s) facing F internally
F_Ex	the fragment(s) facing F externally
F_In_Ex	the fragment(s) facing F_In externally
F_{+i}	the ith neighbor traced from F counter-clockwise
F_{-i}	the ith neighbor traced from F clockwise

indices are used for indexing throughout the fragments lying within the effective region of F. We elaborate their details in Table 2.3. Note that F can also be denoted as F_0. We use F_In to represent F's internally facing fragment, F_Ex meaning the externally facing fragment. Similarly, F_ExIn is the internally facing fragment of F_Ex and F_InEx is the externally facing fragment of F_In. Also for each fragment, we mark its adjacent neighbors with ascending indices in clockwise order. For example, the first adjoining neighbor of F in clockwise order is denoted as F_{+1}, and F can also be denoted as F_0, as shown in Fig. 2.15(b). These indices are used for indexing the fragments lying within the effective region of certain polygon of interest in the context characterization process.

Next, we present the characterized context of fragment F in the format of a 1D data vector defined as in Eqn.(2.6)-(2.7):

$$V_F = \coprod_{i}^{\widetilde{F}_i \in \delta_r^F} \{f_{corn}(\widetilde{F}_i) \oplus f_{ext}(\widetilde{F}_i) \oplus f_{int}(\widetilde{F}_i) \oplus f_{misc}(\widetilde{F}_i)\} \tag{2.6}$$

$$\widetilde{F} = [F, F\_Ex, F\_In, F\_InEx, F\_ExIn...] \tag{2.7}$$

where F is an integer ID number representing a certain fragment in the layout, δ_r^F is the effective region of F. Operators $\bigoplus$ and $\coprod$ are matrix operations (swapping and reordering of elements) for the generation of vector V_F. The length of vector V_F equals the total number of features M.

In this dissertation, the parameter vector V_F formed by the context characterization is defined as the *hotspot signature measurements*, and the

context characterization for each F is also referred as *feature extraction* process. Note the set of all V_F's over the entire layout forms the final output of the *Layout Analyzer*. Our proposed analyzer successfully filters out detection noise and provides a compact data set for *Hotspot Identifiers* (with machine learning models) to be properly established. Meanwhile, using table lookup approaches inside [1] environment, Eqn.(2.6) can be realized with constant time complexity for all fragments in the entire layout via invoking the operators constructed above. This allows us to achieve up to hundreds of times of run-time reduction compared with some previous studies. Simulation results and further discussions will be presented in Subsection 2.3.6.

2.3.4 Hotspot Identifiers with Robust Learning Models

Our *Hotspot Identifiers* employ machine learning models that play essential roles unlike previous works. Using powerful machine learning techniques, we build efficient models specially suitable for classifying lithography hotspot data. In particular, we modify and enhance two types of machine learning techniques: Artificial Neural Network (ANN) and Support Vector Machine (SVM) in mainly two aspects: first, robustness and accuracy in the weight update process; second, detection threshold $\Delta(\cdot)$ optimizations for simultaneous $Hhit$ improvement and $Hextra$ suppression.

Generally speaking, ANN and SVM perform similarly for binary classifications. Although SVM guarantees global optimum in its formulation if the kernel satisfies Mercer's condition, it is usually prone to noise and may also

result in longer run-time for high dimensional data when the number of support vectors is large. ANN on the other hand, provides more noise robustness, compact models and flexible network structures.

With these considerations, we incorporate both classes of models into our *Hotspot Identifiers* with special modifications. For ANNs, we modify the *Resilient Backpropagation* update method [102] with enhanced robustness and better parameter trade-offs between convergence speed and detection accuracy. We also propose strategies for optimizing the detection threshold of each ANN model. For SVMs, we combine a *C-type* SVM formulation with higher accuracy working set selection based on [49], together with the detection threshold optimizations. We describe our special *Hotspot Identifier* model formulations and implementations briefly as follows, with related symbols and variables summarized in Table 2.4.

2.3.4.1 ANN: Artificial Neural Network Models

We present *Hotspot Identifiers* with ANN models, together with the techniques used to configure and apply these novel identifiers.

A typical ANN classifies data by predicting a value for each V_p based on an established set of weights and biases assigned to certain neural network structure. Our ANNs are customized with single hidden layer of neurons, with transfer functions denoted as f_{hid}. Inputs V_p to the ANNs are the extracted feature vector samples labeled with values (y_p) indicating hotspot or non-hotspot patterns (these values can be continuous for variability prediction).

Table 2.4: ANN/SVM related variables

Variables	descriptions
N	total number of input sample vectors
M	feature number per sample vector
V_p	input sample vectors, p=1 to N
V_p^i	the ith element(feature) of V_p, i=1 to M
y_p	hotspot label for V_p in *calibration*, p=1 to N
f_{in}	input transfer function for ANN
f_{hid}	hidden layer transfer functions for ANN
f_{out}	output layer transfer function for ANN
out_p	ANN output prediction value from V_p input
out_{hid}^j	ANN hidden layer jth neuron prediction output
$\overrightarrow{\omega}$	ANN model matrix of neuron connection weight
$K(V_i, V_j)$	SVM kernel function between V_i and V_j
α	SVM weight vector for input V_p's
$\Delta(\cdot)$	threshold function for hotspot decision making
$Est_{\tilde{p}}$	machine learning estimation for a new input $V_{\tilde{p}}$

We use p to represent feature vector index with $p = 1$ to N, V_p^i denotes the ith element of vector V_p, $i = 1$ to M, where M is the total number of features for each sample vector. We use f_{in} and f_{out} to represent input and output layer transfer functions, and index i, j, k to indicate neuron indices in the input, hidden and output layer respectively. In particular, we choose a linear output function, *sigmoid* hidden layer functions and formulate our ANN model calibration process in Eqn.(2.8) to Eqn.(2.15) as follows:

$$objective : minimize\{\sum_{p=1}^{N} E^p\} \quad w.r.t \quad \omega_{ij}, \omega_{jk} \tag{2.8}$$

$$E^p = \frac{1}{2}[out_p - y_p]^2 \tag{2.9}$$

$$out_p = f_{out}\{\sum_j \omega_{jk} \cdot f_{hid}(\sum_i V_p^i \cdot \omega_{ij})\} \tag{2.10}$$

$$\frac{\partial E^p}{\partial \omega_{jk}} = (out_p - y_p) \cdot f_{hid}\{\sum_i V_p^i \cdot \omega_{ij}\} \tag{2.11}$$

$$\frac{\partial E^p}{\partial \omega_{ij}} = (out_p - y_p) \cdot \omega_{jk} \cdot V_p^i \cdot (1 + out_{hid}^j)(1 - out_{hid}^j) \tag{2.12}$$

$$f_{hid} = \frac{2}{(1 + e^{-2x})} - 1, \qquad f_{in} = f_{out} = x \tag{2.13}$$

$$sign\_func(x) = \begin{cases} -1 & \text{x} < 0 \\ 0 & \text{x} = 0 \\ +1 & \text{x} > 0 \end{cases} \tag{2.14}$$

$$Est_{\tilde{p}} = \Delta\{f_{out}[\sum_j \omega_{jk} \cdot f_{hid}(\sum_i V_{\tilde{p}}^i \cdot \omega_{ij})]\} \tag{2.15}$$

As shown in Eqn.(2.8), the objective function is set to the Summed Square Error (SSE) among all N input sample vectors. Such a minimization is achieved through iterative update of weight matrix $\overrightarrow{\omega}$ using modified resilient backprop method. We call every iteration as one epoch, defined as one complete representation of V_1 through V_N to the ANN model. out_p is the ANN prediction result for each V_p vector, while out_{hid}^j is the predicted output from the jth node in the hidden neuron layer, both of out_p and out_{hid}^j are iteratively updated across epoches.

The training and configuration of each *Hotspot Identifier*-ANN model is achieved by executing Algorithm 4, which takes in data set V_p's after the signature extraction process (the *Layout Analyzer*) and returns final neuron weight coefficient matrix $\overrightarrow{\omega}$. In particular there are 3 steps involved for hotspot detection accuracy enhancement: First, the input data set is normalized for

each feature across the whole sample space. Second, the data set is divided into training, validation and testing subsets for the considerations of model robustness, data over-fitting prevention and proper early stopping criteria. In the third step, network output is adjusted incrementally with a step-wise update of network weight matrix towards minimal SSE value with parameters and gradient update procedures modified for hotspot detection under real manufacturing conditions. With the calculations of the gradient values using Eqn.(2.10) to (2.13) and the arithmetic smoothing steps, the *Hotspot Identifier*-ANN is trained iteratively until a certain error target is met.

Algorithm 4 Pseudo-codes for training and configuring *Hotspot Identifier*-ANN( )

Require: Training vectors V_p's and training targets y_p's
  Scale each feature column of the training row vectors to [-1 +1]
  Divide training set into learning set, valid set and test set
  Set converging speed parameters: $\eta_+ = 1.5$, $\eta_- = 0.5$, $\delta_{max} = 50$
  Initialize conditions for all variables to be updated
  while error target not met **do**
    for each V_p in the learning set **do**
      calculate gradients $\partial E^p/\partial \omega_{ij}$ and $\partial E^p/\partial \omega_{jk}$ for V_p
    end for
    calculate the average gradient value for each link as $\partial E^p/\partial \omega_{ij}$
    for all weights and biases **do**
      if $\partial E^p/\partial \omega_{ij}(t-1)$ * $\partial E^p/\partial \omega_{ij}(t) > 0$ **then**
        $sign = \eta_+$
      else if $\partial E^p/\partial \omega_{ij}(t-1)$ * $\partial E^p/\partial \omega_{ij}(t) < 0$ **then**
        $sign = \eta_-$
      else
        $sign = 1.0$
      end if
      $\delta(t) = \min(sign^*\delta(t-1), \delta_{max})$
      $\omega_{ij}(t)$ = -$\delta(t)^*sign\_func(\partial E^p/\partial \omega_{ij}(t))+\omega_{ij}(t-1)$
    end for
    update error for current epoch t
    break if(early stopping criteria met in valid and test sets)
  end while
  return A *Hotspot Identifier* model with ω_{ij},ω_{jk}

Note in Algorithm 4, the training targets are derived by running accurate lithography simulations at a one-time run-time cost usually in the scale of hours. Once the ANN machine learning model is fully trained and configured, we can apply it to identify lithography hotspots according to Algorithm 5 without using costly lithographic simulations.

Algorithm 5 Pseudo-codes for applying *Hotspot Identifier*-ANN( )

Require: V_p's from the *Layout Analyzer*
  Scale the features of V_p's correspondingly by the (*min, max*) values from Algorithm 4
  Load *Hotspot Identifier-ANN* model
  Calculate Equation (2.15)
  return A hotspot estimate $Est_{\tilde{p}}$

2.3.4.2 SVM: Support Vector Machine Models

We present *Hotspot Identifiers* with SVM models, together with the techniques used to configure (Algorithm 6) and apply (Algorithm 7) these novel identifiers.

SVM classifies sample vectors by calculating a (hyperplane) boundary with maximum separation margin in-between of different classes. With such an optimized margin, only the sample vectors forming the boundaries are considered as contributing factors for new sample classifications. These vectors are called support vectors, they are assigned different weights and they perform classification tasks through certain kernel function $K(V_i, V_j)$. For our high fidelity detection flow, we combine a typical 2-class soft-error tolerant SVM, a special working set selection technique using second order information [49] and a detection threshold Δ optimization procedure towards simultaneous accuracy enhancement and false-alarm suppression.

The dual problem of our quadratic formulation of C-type SVM is given as follows in Eqn.(2.16) to Eqn.(2.21):

$$objective : minimize\{f(\alpha) = \frac{1}{2}\alpha^T Q\alpha - e^T\alpha\} \quad w.r.t \quad \alpha \tag{2.16}$$

$$subject \quad to: 0 \leq \alpha_i \leq C, i = 1, ..., N, \tag{2.17}$$

$$y^T \cdot \alpha = 0 \tag{2.18}$$

$$K(V_i, V_j) = exp\{\gamma \cdot \|V_i - V_j\|^2\} \tag{2.19}$$

$$slope\_func(x) = \begin{cases} 0 & \text{x} \leq 0 \\ x & 0 < \text{x} < \text{C} \\ C & \text{x} \geq \text{C} \end{cases} \tag{2.20}$$

$$Est_{\tilde{p}} = \Delta\{\sum_i \alpha_i y_i K(V_{\tilde{p}}, V_i) + bias\} \tag{2.21}$$

Given V_i, i=1 to N sample vectors, with label y_i (either +1 or -1 for 2-class SVM). e is a vector of all 1's. C is a pre-set upper bound to constrain feasible regions for hotspot detection under real manufacturing conditions. Q is N by N positive semi-definite matrix defined as $Q_{ij} = y_i y_j K(V_i, V_j)$, where $K(V_i, V_j)$ is defined in Eqn.(2.19) as the kernel function. α is the N element weight vector for V_p's. Note α is generally sparse and the non-zero weights correspond to the final support vectors. Due to the fact that Q is usually dense and large, decomposition methods are usually used to solve the formulation iteratively rather than directly dealing with the quadratic Eqn.(2.16).

The training and configuration of *Hotspot Identifier*-SVM models are achieved through performing Algorithm 6, which intakes data set V_p's and returns the supporting vectors and corresponding weight coefficients. There are 3 major steps involved: First, data set normalization for detection robustness; Second, high order working set selection for enhanced detection accuracy particularly for our special hotspot detection requirements; Third, update weight and gradient vectors. The last 2 steps are carried out in an iterative manner

Algorithm 6 Pseudo-codes for training and configuring *Hotspot Identifier*-SVM( )

Require: training vectors V_p's and training targets y_p's
  Scale each feature column of the training row vectors to [-1 +1]
  Set control parameters: γ = -1/M, C = 1.5, stopping tolerance ϵ = 1e-3, min floating number τ = 1e-12
  Initialize weight vector α and gradient vector G
  while 1 **do**
    Working set (i, j pair) selection based on [49]
    if j == -1 **then**
      break
    end if
    Calculate $\eta_1 = \max(\ \tau,\ Q_{i,i} + Q_{j,j}$ - 2 $y_i\ y_j\ Q_{i,j})$
    Calculate $\eta_2 = y_j \cdot G_j$ - $y_i \cdot G_i$
    Update weight: ϖ_i += $y_i \cdot \eta_1$ / η_2, ϖ_j -= $y_j \cdot \eta_1$ / η_2
    $\varpi_i = slop\_func(\varpi_i)$,$\varpi_j = slop\_func(\varpi_j)$
    Update gradients for all k = 1 to M: G_k += $Q_{k,i}$ (ϖ_i - ϖ_i^{prev}) + $Q_{k,j}$ (ϖ_j - ϖ_j^{prev})
  end while
  Calculate ρ and *bias* values for prediction processes
  return A *Hotspot Identifier* model of non-zero elements of α and corresponding V_p's

until certain error target is met. For implementation details regarding the higher order working set selection please refer to [49]. When the SVM model is fully trained and configured, we can apply it to evaluate a new design pattern using Algorithm 7. Note in Algorithm 7, there is no need for accurate lithography simulations.

To sum up, our *Hotspot Identifiers* with modified ANNs and SVMs hold their respective advantages as two of the most important machine learning classifiers, which are fine-tuned for our hotspot detection requirements under

Algorithm 7 Pseudo-codes for applying *Hotspot Identifier*-SVM( )

Require: V_p's from the *Layout Analyzer*
  Scale the features of V_p's correspondingly with the (*min, max*) values from Algorithm 6
  Load *Hotspot Identifier-SVM* model
  Calculate Equation (2.21)
  return A hotspot estimate $Est_{\tilde{p}}$

real manufacturing conditions. From a data mining point of view, evaluating both types of classifiers can allow us deeper insights into the interpretation of the dataset. Further observations and discussions will be offered in Subsection 2.3.6.

2.3.5 Integrative Flow for Successive Identification Refinements

2.3.5.1 A Quick Overview

Due to the small number of real hotspots and the highly noisy detection environment under real manufacturing conditions, we propose a novel successive flow with successive levels of refinements to integrate our proposed fast *Layout Analyzer* and powerful *Hotspot Identifiers*. In its nature, such an approach hybrids the strength of *Hotspot Identifier* models and the successive levels of pattern classifications, contributing to significant detection performance boost in run-time, accuracies and false alarms, when compared with previous approaches such as straight-forward machine learning.

Following our previous discussions, we present the pseudo codes in Algorithm 8 and Algorithm 9 for the *Calibration Stage* and the *Detection Stage* overviewed in Subsection 2.3.1. For both algorithms, the proposed hierarchy

Algorithm 8 Pseudo-codes for the *Calibration Stage*

Require: A small set of input design layouts
  Setup optical models, fragmentation specifications
  Generate training targets by accurate lithographic simulation
  Invoke the *Layout Analyzer*
  for each fragment in the design layout **do**
    Perform *Hotspot Signature Measurements*
    Update Look-up tables
  end for
  Invoke ***Hotspot Identifier*-ANNs**(or **-SVMs**)(*signatures*, *targets*) for supervised learning processes with successive refinements (Fig. 2.16)
  return Compact *Hotspot Identifiers* models

Algorithm 9 Pseudo-codes for the *Detection Stage*

Require: A large set of new input layouts
  Setup (the same) optical models, fragmentation, etc.
  Invoke the *Layout Analyzer*
  Load the *Hotspot Identifiers*
  for each fragment in the design layout **do**
    Apply the *Hotspot Identifiers* inside a novel *Successive Refinement Hierarchy* (Fig. 2.17)
  end for
  return Identified hotspot patterns

of successive identification refinements plays a critical role in both detection accuracy and run-time. Such a hierarchy takes slightly different forms in the calibration than in the detection stage. As shown in Fig. 2.16, in the *Calibration Stage* the hierarchy takes the form of a multi-level cascade with each level contributing an unique *Hotspot Identifier* model. As illustrated in Fig. 2.17, various successive levels of *Hotspot Identifiers* derived from the *Calibration Stage* are applied in the *Detection Stage* in a similar successively refined manner to help reduce the false alarm rate $Hextra$ without penalizing the hotspot

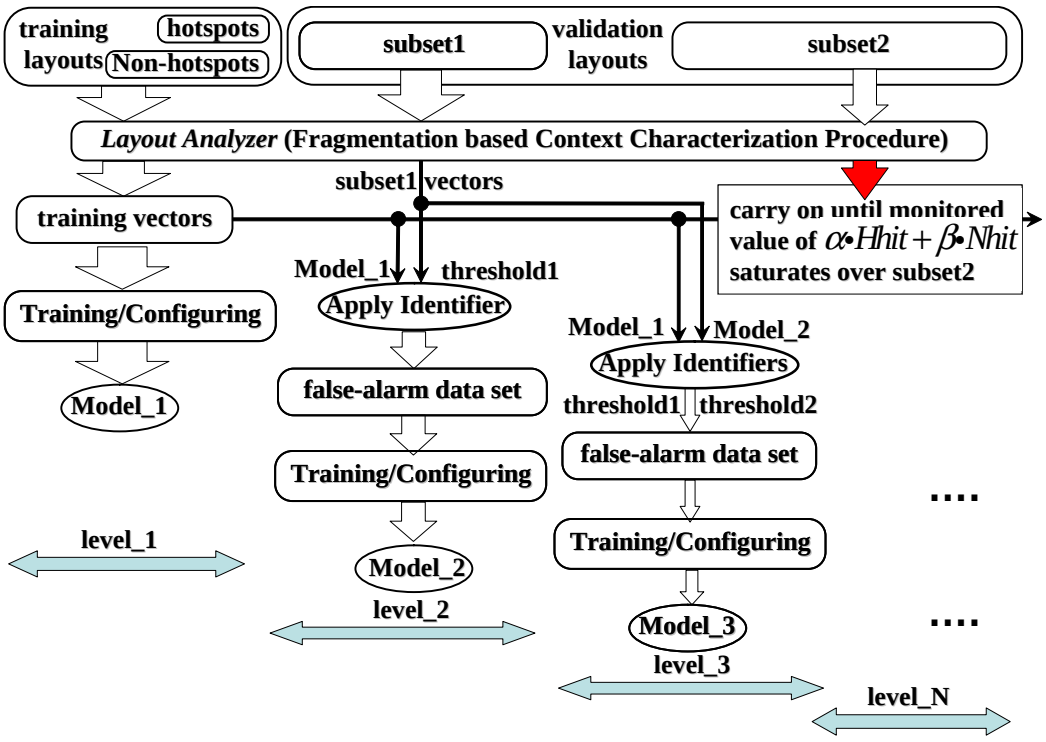


Figure 2.16: Configuring successive *Hotspot Identifiers* and thresholds

detection rate $Hhit$. In the following subsections, we explain such a hierarchy in detail by dividing it into two terms: a *Global* and a *Local* term.

2.3.5.2 Global Calibration and Detection

Here we refer to the first level training in Fig. 2.16 as the global training, since *Hotspot Identifier* $Model\_1$ (abbreviated as $Model\_1$) is trained with the whole training dataset (on the global scale); similarly in Fig. 2.17, detection with only $Model\_1$ is defined as global detection, since the whole testing data go through this model. As we will show later in Subsection 2.3.6, using the

Layout Analyzer and *Hotspot Identifiers*, the global term alone achieves very satisfactory hotspot detection accuracy $Hhit$ but low non-hotspot detection accuracy $Nhit$. Under the evolving manufacturing conditions, hotspot and non-hotspot patterns are highly unbalanced in quantity, which results in huge number of non-hotspot patterns. Under such a scenario, even a small fraction of identification error can lead to highly excessive false alarms therefore heavy workload of post-design hotspot removal. Consequently, *Global* term alone is not enough to ensure overall satisfactory detection performance.

2.3.5.3 Successive Local Refinements

To further suppress false alarms meanwhile maintaining satisfactory hotspot detection rate, we extend the *Global* term with sub-levels of identification hierarchies, which we refer to as the *Successive Local Refinements*. As illustrated in Fig. 2.16 and Fig. 2.17, we apply the *level_2* to *level_N Hotspot Identifiers* that serve as successive stages of refinements in both *calibration* and *prediction* stages. In the *Calibration Stage*, the refinement flow consists of several key steps: (1) training, configuring and validating multiple *Hotspot Identifiers* using the entire training data set plus the false alarm data sets accumulated with each additional level. (2) stopping criteria to decide when to stop adding more *Hotspot Identifiers*. (3) optimizations of the thresholds associated with the *Hotspot Identifers*. In the *Detection Stage*, all the *Hotspot Identifier* models and thresholds are applied successively, and hotspots are detected as those patterns that are eventually identified as 'hotspots' after all

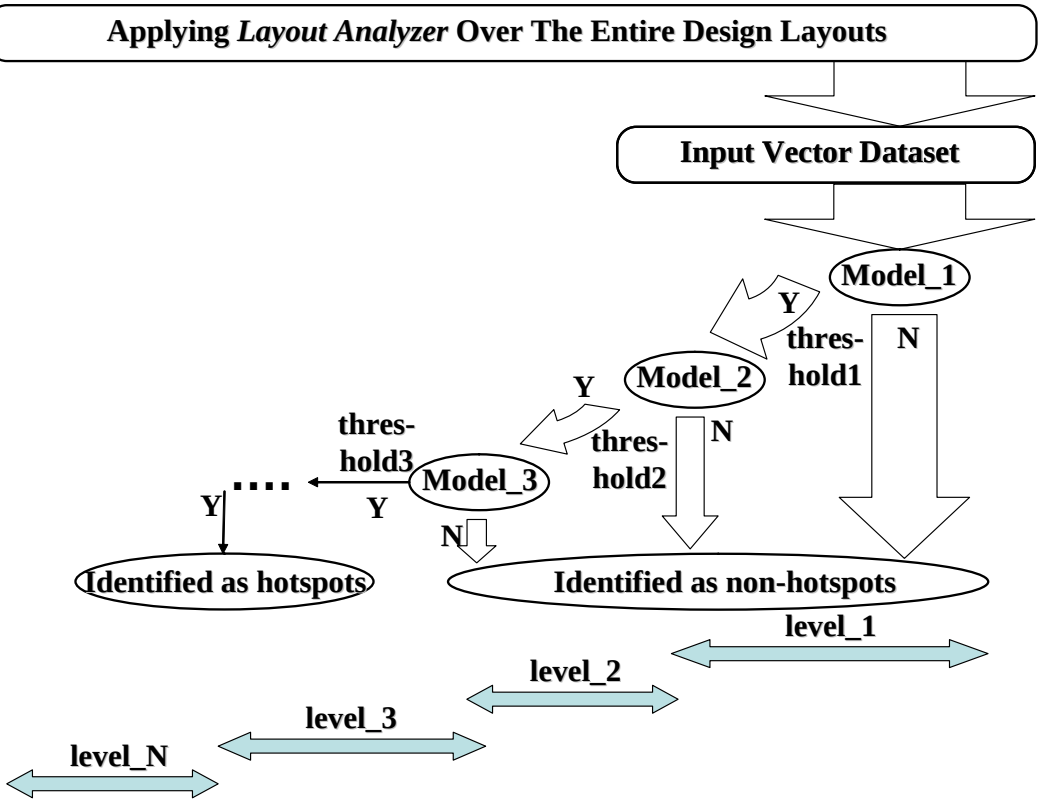


Figure 2.17: Applying successive *Hotspot Identifiers* and thresholds

levels of refinements. We describe the key steps in more detail as follows:

Configuring Successive Hotspot Identifiers

The configuration of each *Hotspot Identifier* involves the training and validation of both the learning model and a detection threshold above which a pattern is identified as a hotspot. Fig. 2.18 shows some motivations on the importance of threshold selection. By now we have a *Hotspot Identifier* $Model\_1$ derived from *Global* stage using Algorithm 4, Algorithm 6 and the training layouts. Then we apply $Model\_1$ (using Algorithm 5, Algorithm 7)

over some validation data *subset1* to derive a $threshold1$. With this threshold, we collect all the identification false alarms (the configuration mistakes) and use them to configure a $Model\_2$. Subsequently, an extra *Hotspot Identifier* $Model\_3$ can be derived over another false alarm set of data generated by applying $Model\_1$ and $Model\_2$ successively (as in Fig. 2.17). Consequently, $threshold2$ can be derived. Such a refinement goes on until our predefined performance metric saturates over the validation data set $subset2$.

Stopping Criteria

To quantify the stopping criteria for the *Local Successive Refinements*, we introduce a user defined performance metric Ψ_{perf} in Eqn.(2.22):

$$\Psi_{perf} = \alpha \cdot Hhit + \beta \cdot Nhit \tag{2.22}$$

where α and β are user defined weights, $Hhit$ is the hotspot detection accuracy and $Nhit$ is the non-hotspot detection accuracy. Therefore, Ψ_{perf} represents the weighed summation of hotspot and non-hotspot detection accuracies. With each additional level we re-evaluate Ψ_{perf} over $subset2$ using all the *Hotspot Identifiers* derived sofar, according to Algorithm 5, Algorithm 7 and Fig.2.17. We stop configuring additional *Hotspot Identifiers* when Ψ_{perf} saturates or starts to degrade.

2.3.5.4 Threshold optimizations

The important role of threshold optimization has been illustrated in Fig. 2.18. Here we employ a heuristic approach, that is to exhaust the solution

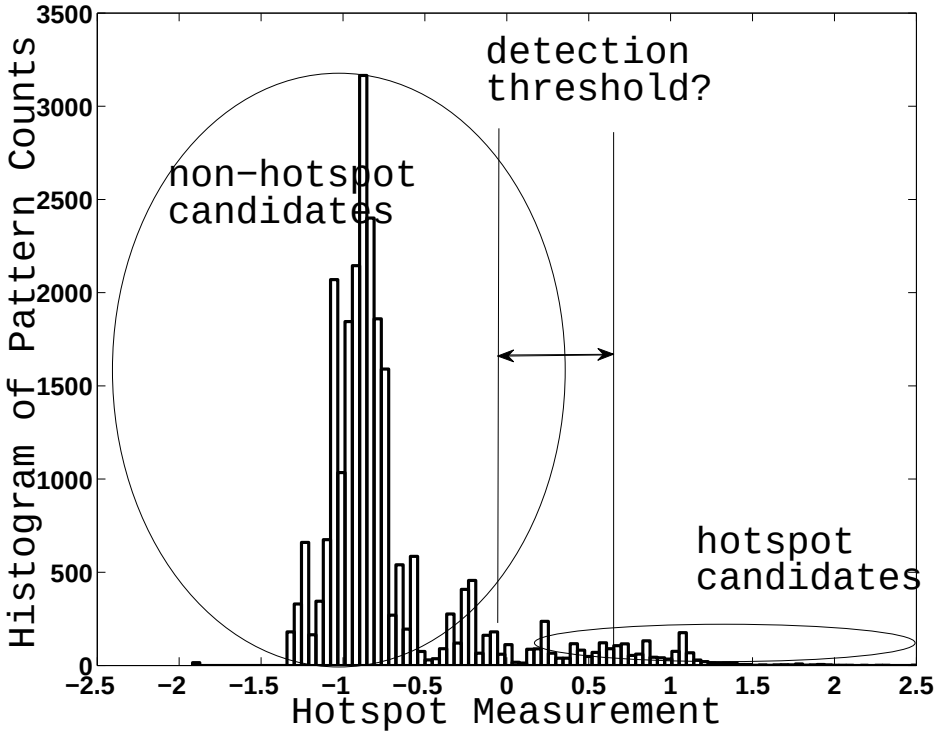


Figure 2.18: Threshold function optimization in hotspot identification

space with grid based simulations and select the threshold combinations giving the best Ψ_{perf}. The main justification is two fold: (1) the *Calibration Stage* is performed only once *a priori* therefore does not lead to run-time overhead for the *Detection Stage*; second, based on our experiments in Subsection 2.3.6, Ψ_{perf} saturates at level 3, therefore the total solution space is limited and the heuristic approach would suffice. For more details of the simulation results please refer to Subsection 2.3.6.

2.3.5.5 Detection Validation and Testing

Testing is carried out over a new set of layouts using the *Layout Analyzer*, the *Hotspot Identifiers* and the refinement architecture in Fig. 2.17. Our proposed flow differentiates the data streams and guides them through successive levels of identifications, resulting in satisfactory overall performance.

2.3.6 Simulation and Testing

The simulation process involves the *Calibration Stage* and the *Detection Stage*. We further break down these 2 stages into 3 steps: (1) under real manufacturing conditions, a *Hotspot Identifier* is trained and configured on a 500 um^2 training layout with fully placed and routed metal tracks in $45nm$ technology (in the *global* stage); (2) the *Local Successive Refinements* and the threshold optimization are carried out over the validation data set; (3) the *Detection Stage* is tested and evaluated on $45nm$ industry design layouts in terms of $Hhit$, $Hmis$, $Hextra$, $Nhit$ and run-time under a real set of industry-strength $45nm$ process manufacturing conditions.

In Fig. 2.19, we plot a fine-grid simulation result of the threshold optimization process in the *Calibration Stage*. The x axis is the hotspot detection accuracy $Hhit$ over the validation set *subset2*, and y axis is the non-hotspot detection accuracy $Nhit$. Every point on the plot represents a different combination of the thresholds as in Fig. 2.16, the data depicted with cross markers is derived through applying *Hotspot Identifier*-ANNs and the circles through SVMs. In this case, ANNs give more noise robustness, while the SVMs achieve

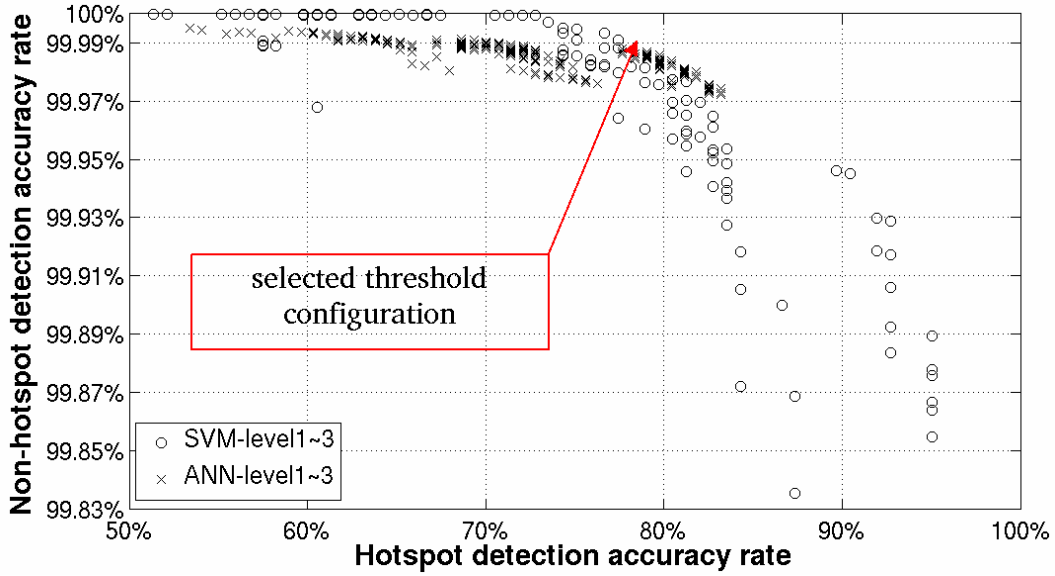


Figure 2.19: Detection accuracies versus the threshold configuration

better performance ψ_{perf}. We pick the upper-right corner combinations on ANN and SVM curves respectively in Fig. 2.19 for the purpose of generating the optimal hotspot identifiers : $Model\_1$ to $Model\_3$.

With the 3 *Hotspot Identifiers* ready, we execute the *Detection Stage* to perform hotspot identifications over 5 industry design layouts under the real manufacturing conditions. Details of layouts C1 to C5 are in Table 2.5, where the sizes of the test cases range from 900 um^2 to 1 mm^2. We can tell from the total number of geometry fragmentation in C1-C5 (hotspot + non-hotspot counts) that they contain very densely placed and routed metal layers.

Table 2.6 shows the simulation results of hotspot identification accuracies using our proposed methodology, where **GD** represents the *Global Detection* stage alone (using only *Hotspot Identifier* $Model\_1$) and **GD+LR**

Table 2.5: Details of testing design layouts

	dimension	hotspot count	non-hotspot count
C1	30 X 30 um^2	4	4.955k
C2	50 X 50 um^2	0	17.37k
C3	200 X 200 um^2	6	293.5k
C4	500 X 500 um^2	38	1779k
C5	1000 X 1000 um^2	137	7175k

Table 2.6: Simulation results of our proposed methodology on industry layouts under real manufacturing conditions

	Using *Hotspot Identifiers*-ANN				Using *Hotspot Identifiers*-SVM			
	GD		GD+LR		GD		GD+LR	
	Hhit[a]	Nmis	Hhit	Nmis	Hhit	Nmis	Hhit	Nmis
C1	4	315	3	18	4	251	4	5
C2	-	109	-	11	-	81	-	3
C3	6	493	5	31	6	355	5	7
C4	32	3020	30	195	34	1983	31	38
C5	121	10960	111	485	122	7535	114	135

[a] Hhit is the number of correctly identified hotspots; Nmis is the number of incorrectly identified non-hotspots (false alarms)

represents the combination of the *Global Detection* and the *Successive Local Refinements* (using $Model\_1$-$Model\_3$).

We observe that although **GD** leads to satisfactory hotspot identification accuracy, it is **LR** that plays a vital role in bringing down the false alarms. Also in these results we better appreciate the detection challenges under real manufacturing conditions due to highly unbalanced quantities of hotspot and non-hotspot patterns. Fig. 2.20 provides a good visualization of detection false alarms on the 1 mm^2 area design C5 by using 2 previous methods and our proposed methodology. From the figure it is obvious that our approach

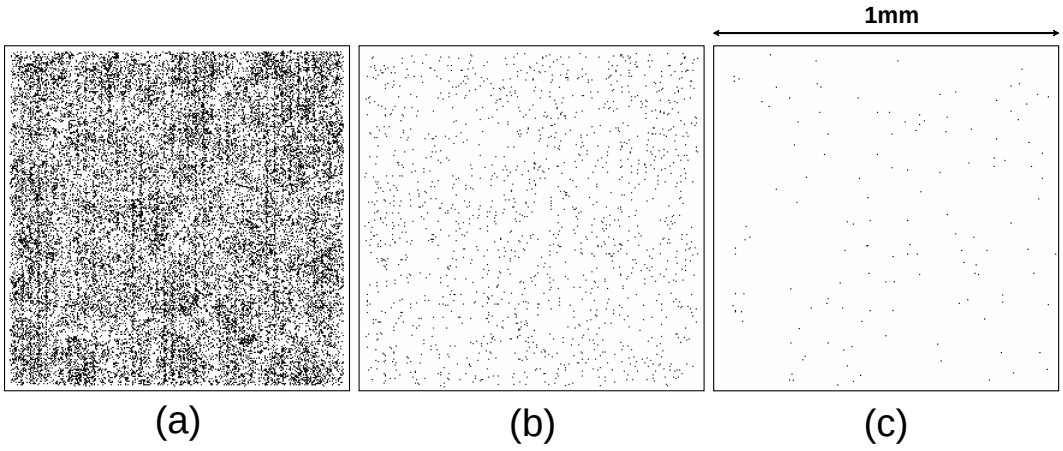


Figure 2.20: Visualizations of false alarm locations when simulating (a) [43] on C5, (b) [120] on C5, and (c) our method on C5 (barely visible:∼100 false alarm spots on $1mm^2$ layout)

achieves the least amount of false alarms (without penalties on the hotspot accuracy $Hhit$), therefore the workload overheads can be kept minimum in the post-layout hotspot removal stages. We can also appreciate that although a 95% of non-hotspot identification accuracy seems good, we have to raise it to above e.g., 99.85% to avoid excessive post-layout corrections.

Table 2.7 and Fig. 2.21 show comparisons of accuracies and run-time between our approach and some existing machine learning based studies. The simulation results of [43] and [120] are collected by directly running the original source codes on our testing cases C1-C5. We also implement the original method of [47] inside [1] via an Advanced Programming Interface in C/C++. Due to environment compatibility issues, we modified the original approach in [47] slightly for better memory efficiency, which could possibly end up with more run-time. However, it would suffice as a first order estimation.

Table 2.7: Performance comparison between previous hotspot identification methods and our method

	DAC09 [47][a]	SPIE09 [120][b]	ICICDT09 [43][b]	Identifiers - ANN		Identifiers - SVM	
				GD	GD+LR	GD	GD+LR
Avg hotspot detection accuracy $Hhit$	88%	80%	87%	88%	82%	89%	83%
Avg Nonhotspot detection accuracy	94.523%	99.985%	99.809%	99.847%	99.994%	99.895%	99.998%
Avg false alarm count per mm^2	300K	1.1K	13.5K	10K	0.45K	7.5K	0.13K
Avg CPU run-time[c] per mm^2	356	30	10	1.5	1.5	2.0	2.0
Avg real-time run-time[d] per mm^2	100	8.50	2.80	0.40	0.40	0.52	0.52

[a] Implemented within the same geometry engine framework [1] with slight modifications for compatibility reasons. Calibrated on a total 100X100 um^2 region from layout C5 due to run-time constraint.

[b] Results collected by running the original source codes on C1-C5

[c] Run-time calibrated in the unit of $\underline{\text{CPU hour}/mm^2}$ on Linux station with 2.8GHz quad-core processors.

[d] Run-time calibrated in the unit of real time $\underline{\text{hour}/mm^2}$ on Linux station with 2.8GHz quad-core processors.

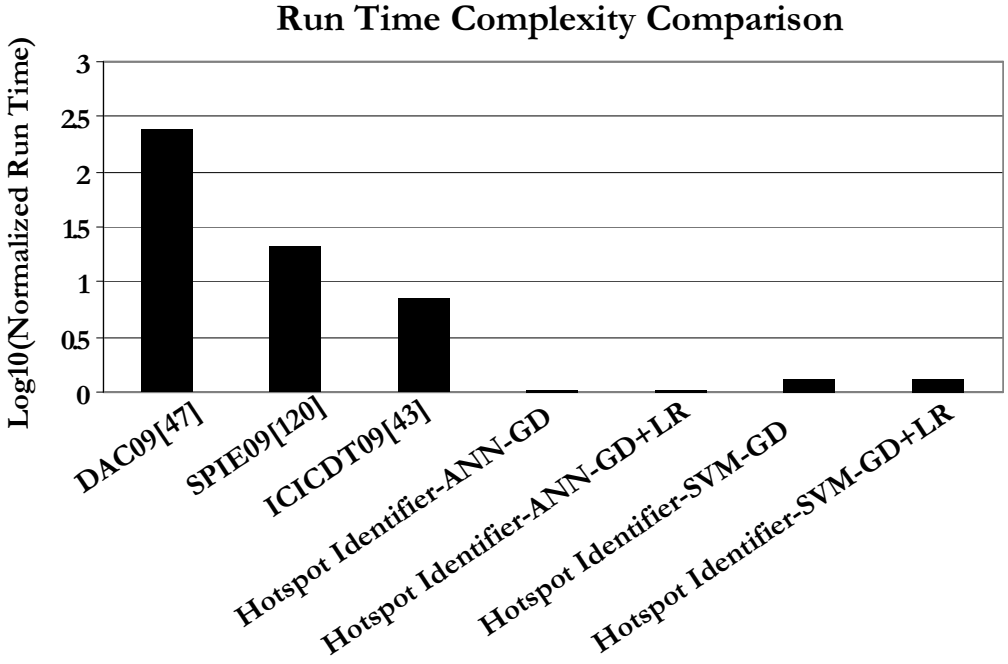


Figure 2.21: Run-time comparison between our approach and previously existing works, in the unit of $Log10(hour/mm^2)$

As shown in Table 2.7, our proposed methods demonstrate better performance with superior run-time under real manufacturing conditions. With similar or slightly better hotspot detection rate $Hhit$ of 82%-89%, we achieve hotspot false alarm reductions ranging from 2.4X (between [120] and *Hotspot Identifiers*-ANN-**GD+LR**) to 2300X (between [47] and *Hotspot Identifiers*-SVM-**GD+LR**). As visualized in Fig. 2.21, the simulation run-time speed-ups range from 5X (between [43] and *Hotspot Identifiers*-SVM) to 237X (between [47] and *Hotspot Identifiers*-ANN), when calibred in *CPU Hour/mm*^{2} unit. Such speedups mainly owe to our proposed identification methodology that is free of time-consuming data transformations, such as grid density extraction,

distance transform and histogram calculations, etc.

Inside our locally refined detection methodology, ANN models result in faster runtime than SVM models while SVM models outperform ANN models in both hotspot and non-hotspot detection accuracy. We also notice that the run-time overhead introduced by the *Successive Local Refinements* is negligible, owing to: (1) the ultra-fast speed of the *Layout Analyzer* and the *Hotspot Identifiers*; (2) the exponential reduction of false alarms achieved by each additional level of refinement. These make our methodology especially suitable for guiding lithography friendly physical design under the real manufacturing condition challenges.

To evaluate the run-time scalability of our methodology on multi-core platforms, we implemented the *Layout Analyzer* and the *Hotspot Identifiers* inside [1] with parallel processing friendly functions and procedures. Assisted by the layout segmentation and refactoring features provided by [1], we plot in Fig. 2.22 the run-time of our approach when simulated on a Linux Workstation with a Intel quad-core processor. In Fig. 2.22, C1-C5 have increasing areas from 900 um^2 to 1.0 mm^2. We can see that our methodology shows linear run-time complexity as design layouts scale to full-chip size. In comparing the CPU time and the real time of both *Hotspot Identifiers*-ANN and *Hotspot Identifiers*-SVM, we observe that our methodology demonstrates very good multi-core scalability. From 'CPU Time' to 'Real Time', ANN models and SVM models achieve 73.3% and 74.0% run-time reduction on quad-core machines, respectively. The main reasons of such complexity and scalability owe

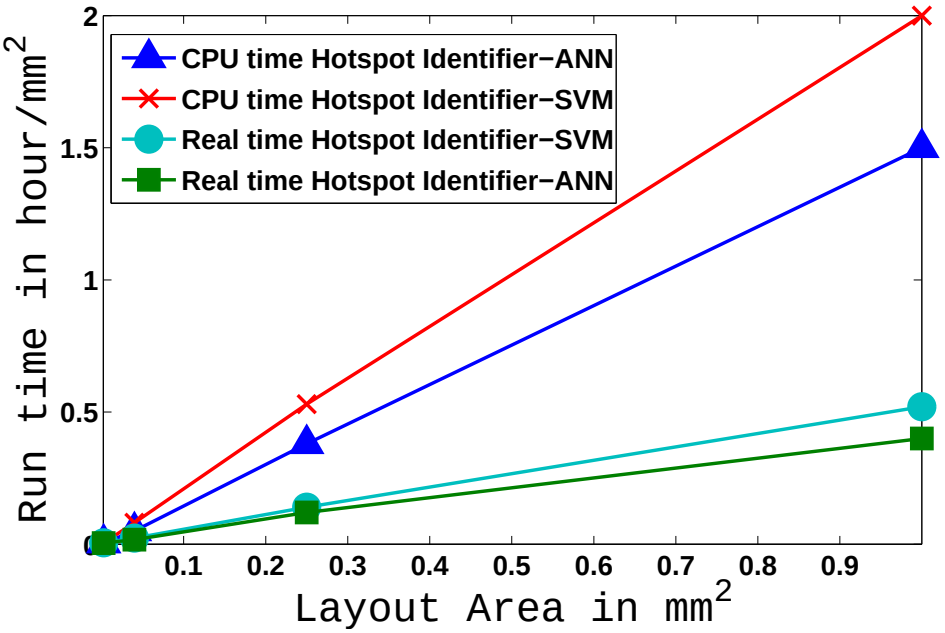


Figure 2.22: The linear run-time scalability of our proposed methodology

to: (1) good memory/workload management and database refactoring mechanisms of [1]; (2) the novel measurement operators we propose in the *Layout Analyzer*; (3) the ultra-fast calculations involved with the *Hotspot Identifiers*.

Based on the results in Table 2.7, we summarize and compare several classes of machine learning based hotspot identification methods in Table 2.8, where we further highlight the key contributions of our proposed methodology. First, we employed powerful *Hotspot Identifiers* to achieve high hotspot detection accuracy. Second, we proposed efficient successive refinements to suppress detection false alarms. Third, we employed ultra-fast *Layout Analyzer* for advantageous run-time efficiency. Without sliding windows or raster scanning,

Table 2.8: Comparisons between existing methods

	[47]	[120]	[43]	**ours**
Identifier model	SVM	regression	ANN	**ANN,SVM**
accuracy	high	medium	high	**high**
false alarms	high	low	high	**low**
window-based	yes	no	yes	**no**
scanning coverage	trade-off	full	trade-off	**full**
run-time	usually slow[a]	medium	medium[a]	**very fast**

[a] For scanning window based approaches , run-time can be traded-off at the cost of detection accuracy and coverage.

our methodology enables full design layout analysis without leaving any cold spots. Consequently there is no need to seek trade-offs between run-time and detection accuracy/coverage. It is obvious that our method is most suitable to guide lithography-aware physical design due to these performance advantages.

Although the machine learning methodology still misses some hotspots, it is acceptable as a fast and high-fidelity prediction at early (physical) design stages. If needed, the small number of missed hotspots could be captured and complemented using pattern matching algorithms without the enumeration issues that pure pattern matching methods suffer.

2.3.7 Summary

Under real and continuously improving manufacturing conditions, lithographic hotspot detection faces many critical challenges. First, real hotspots become hard to detect at early design cycles and hard to fix at post-layout stages. Second, false alarm rate must be kept minimal to avoid excessive and

expensive post-processing hotspot correction. Third, full chip physical verification and optimization require fast turn-around time. Last but not least, constantly evolving RETs and manufacturing conditions favor generic hotspot identification methodologies.

To alleviate the huge run-time cost of current lithographic hotspot simulators meanwhile addressing the above issues, we proposed an ultra-fast and high fidelity hotspot detection methodology providing full layout, feature-centric assessment as improvement over sliding window/raster scanning and sampling-based techniques. Under the real manufacturing conditions, we incorporated a novel set of *hotspot signature measurements*, powerful *Hotspot Identifiers* with machine learning models and a successively refined detection flow. We implemented and tested our algorithms with an industry-strength engine [1] under industry-strength PDK/manufacturing conditions, and showed that it significantly outperforms previous state-of-the-art algorithms in hotspot detection false alarm rate (2.4X to 2300X reduction) and simulation run-time (5X to 237X reduction), with similar or slightly better hotspot detection accuracies. The demonstrated high performance makes our approach very suitable for identifying lithography hotspots in the early design stages and guiding lithography-friendly physical design at run time. Potential combinations with existing pattern matching techniques can possibly deliver even more powerful CAD environments for the ultimate detection accuracies.

2.4 Ultra-High Performance Hotspot Detection Using Meta-Classification Methodology

Due to the widening gap between the continuous scaling of feature-size and the limited lithography capability[8], the semiconductor industry is more and more critically challenged in both design and manufacturing domains. To properly address these challenges, various Design-Aware Manufacturing and Manufacturing-Aware Design techniques have been developed to eliminate lithography hotspots and to ensure high product yield at post-Si stage.

In the manufacturing domain, powerful resolution enhancement techniques (RET) have been proposed for mask re-targeting and optimization, such as Sub-Resolution Assist Feature insertion (SRAF), Optical Proximity Correction (OPC) and Double Patterning Lithography (DPL) (e.g., LELE-DPL and SADPL). With these techniques, the masks of a design are optimized for printability and the lithography hotspots are located/fixed in a construct-by-correction manner. In the design domain, CAD methodologies have evolved to incorporate RET models into early design stages (e.g., detailed routing) and to avoid lithography-unfriendly patterns in a correct-by-construction manner [27, 34, 39, 89, 97]. For applications in both domains, fast and high fidelity detection methodologies are highly demanded to locate process hotspots accurately. Such techniques can also greatly benefit physical verifications and design rules development, etc.

However, the quests for such detection methods are critically challenged in many aspects: (1) designs are getting exponentially more complex; (2) under

the evolving manufacturing conditions, the number of real hotspots is but a very small fraction of the entire design, making it very difficult to achieve high detection accuracies and low false-alarms simultaneously; (3) detections are seriously run-time constrained due to short turn-around-time, etc.

Current state-of-the-art hotspot detection methods mainly fall into the following 3 categories with their respective strengths and drawbacks. (1) Methods that utilize accurate lithography simulations [67, 103] achieve 100% detection accuracies and 0% false-alarms, but their huge run-time makes them infeasible for most applications. (2) Methods that employ machine learning techniques detect hotspots via the training and application of a data classifier[42, 43, 47, 92, 132, 133]. Among recently demonstrated works, various types of classifiers have been explored and implemented, including Artificial Neural Network (ANN) models, Support Vector Machine (SVM) models and sequential/hierarchical refinements using several models back-to-back. These works have shown good false-alarm rate and noise suppression capabilities, however the detection accuracies are not very satisfactory. (3) Methods that use pattern matching techniques[66, 83, 135, 138] search for certain patterns on a pre-defined hotspot blacklist. Once hotspot patterns are well-defined, such methods can operate with 100% detection accuracies. However, the major drawback lies in the pattern enumeration process, especially when new design layouts/patterns are involved after the patten library is built. In practice, pattern matchers are designed to perform fuzzy/flexible matchings, which in turn significantly increases the detection false-alarms.

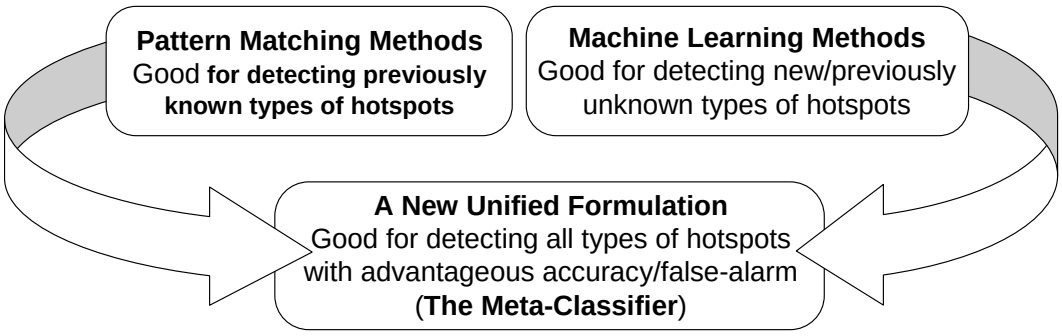


Figure 2.23: A unified meta-classifier to combine the strengths of various detection techniques (e.g., machine learning and pattern matching)

Recently in [131], a hotspot detection flow is proposed to hybrid the strengths of machine learning models and pattern matching models. Such a flow feeds data samples to a pattern matcher first, then employs machine learning classifiers to further examine the non-hotspot data set produced by the pattern matcher. The flow demonstrates better trade-offs between detection accuracies and false-alarms suppression comparing with previous works. However, the ad-hoc nature of such a flow does not guarantee the optimal solution and it could be very sensitive to the noise of each model involved.

In order to better address the issues above, we propose *EPIC*: an efficient meta-classification formulation to combine various existing hotspot detection techniques into a unified framework which selectively adopts the strengths and suppresses the drawbacks of each technique. Based on the theoretical grounds proposed in [122], we developed a new CAD flow with different types of base classifiers and configure the flow optimally via constrained quadratic programming techniques.

The rest of the section is organized as follows. In Subsection 2.4.1 we further motivate the *Meta-Classification* methodology and summarize our main contributions. Subsection 2.4.2 details the construction of the *Meta-Classifier* together with the overall CAD flow used to calibrate and apply the classifiers. Subsection 2.4.3 describes the mathematical formulations and techniques developed to optimize the *Meta-Classifier*. Subsection 2.4.4 discusses the configurations of *Base Classifiers*, followed by Subsection 2.4.5 presenting the simulation results and performance analysis/comparions. Subsection 2.4.6 serves as a brief summary of this section.

2.4.1 Motivation and Contribution

With Fig. 2.23 we have discussed the need for a systematic and unified meta-classification methodology to selectively combine certain features of multiple hotspot detection engines. We further motivate such a meta-classifier with examples shown in Fig. 2.24.

Fig. 2.24 presents the printed images of 2 local regions from certain design at $32nm$ technology node after applying RETs. We can make several observations from Fig. 2.24(a) and (b). First, there are various types of process hotspots, featuring special patterns related to line-ends, jogs, corners or contacts, etc. Second, all patterns in the design suffer from process variations, but at certain different degree. Therefore, depending on EPE threshold, we can categorize process hotspots into multiple classes. This way we can more effectively study the effects that each hotspot class has with respect to

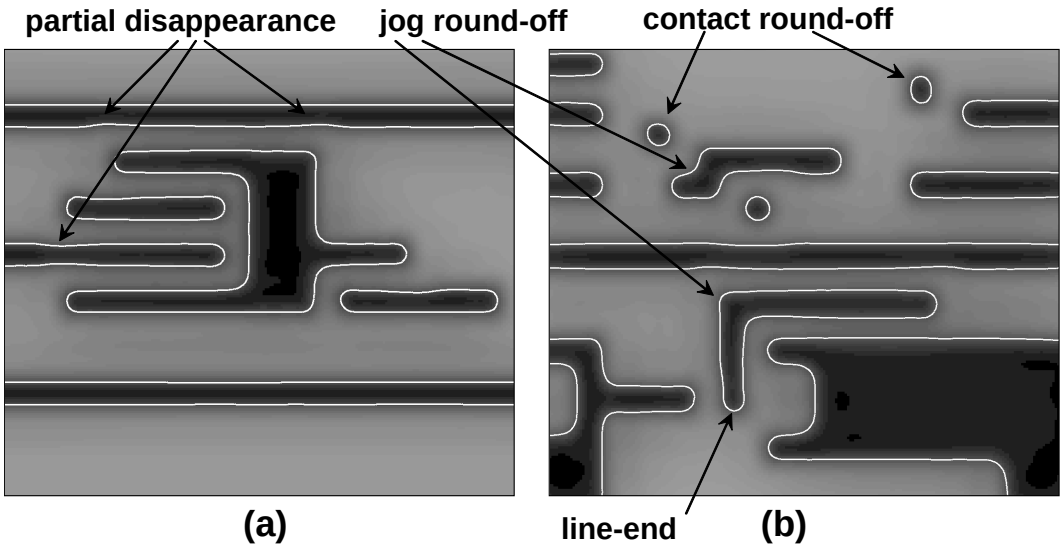


Figure 2.24: A motivational example for the challenges of high performance process hotspot detection

manufacturability and yield.

During the meta-classification process, each fragment geometry in the layout will be examined by multiple hotspot detection engines. Suppose the ith fragment is assessed by a machine learning classifier ML and a pattern matcher PM as the same time and the detection results are x_i^{ML} and x_i^{PM}, respectively. x_i^{ML} takes certain value between -1 (non-hotspot) and +1 (hotspot), while x_i^{PM} usually is either -1 (non-hotspot) or +1 (hotspot). Thus the meta-classification problem becomes equivalent to the following problem:

Problem: *Given* x_i^{ML} and x_i^{PM}, *decide* the final hotspot prediction lable T_i^{meta} of the meta-classifier to further improve detection performance.

First, it is easy to see that T_i^{meta} is +1 if both x_i^{ML} and x_i^{PM} are +1;

T_i^{meta} is -1 if both x_i^{PM} and x_i^{ML} are -1. Second, in the cases when x_i^{ML} and x_i^{PM} disagrees with each other, we introduce the *Mapping Functions* $f^{ML}(\cdot)$ and $f^{PM}(\cdot)$ to adjust the weights and control the detection performance.

$$T_i^{meta} = F^{meta}\{x_i^{ML} \cdot f^{ML}(x_i^{ML}) + x_i^{PM} \cdot f^{PM}(x_i^{PM})\} \tag{2.23}$$

If we define T_i^{meta} as in Equation (2.23) above, we can pre-calibrate the *Mapping Functions* with accurate lithography simulations as golden targets. Then we can use the calibrated functions onto new layout fragments by applying Equation (2.23). F^{meta} is a threshold cut-off function defined as follows,

$$F^{meta}(x) = \begin{cases} +1(hotspot), & \text{if } x \geq \theta; \\ -1(nonhotspot), & \text{if } x < \theta. \end{cases} \tag{2.24}$$

Such a formulation is generic in the sense that if $f^{ML}(x)$=1 and $f^{PM}(x)$=0, then Equation (2.23) degenerates into a machine learning classifier ML; similarly for pattern matcher PM. Therefore the performance optimization of a meta-classifier becomes the optimization of the *Mapping Functions*. Base on such a motivation, we proposes a systematic optimization flow to construct the optimal meta-classifiers with N disparate detection engines, known as the *Base Classifiers*. We summarize the key contributions of this section as follows,

- We propose for the first time a unified meta-classifier to seamlessly combine the advantages of various types of hotspot detection techniques without time-consuming lithography simulations in detection stage.
- We develop high performance hotspot detection engines as *Base Classifiers* to leverage current state-of-the-art machine learning and pattern matching techniques.

- We employ Quadratic Programming techniques to achieve optimal performance optimization of the meta-classifier.

- We perform exhaustive assessment on the proposed method using various industry-strength benchmarks under advanced RET and manufacturing conditions.

2.4.2 Meta-Classification Methodology and Overall Flow

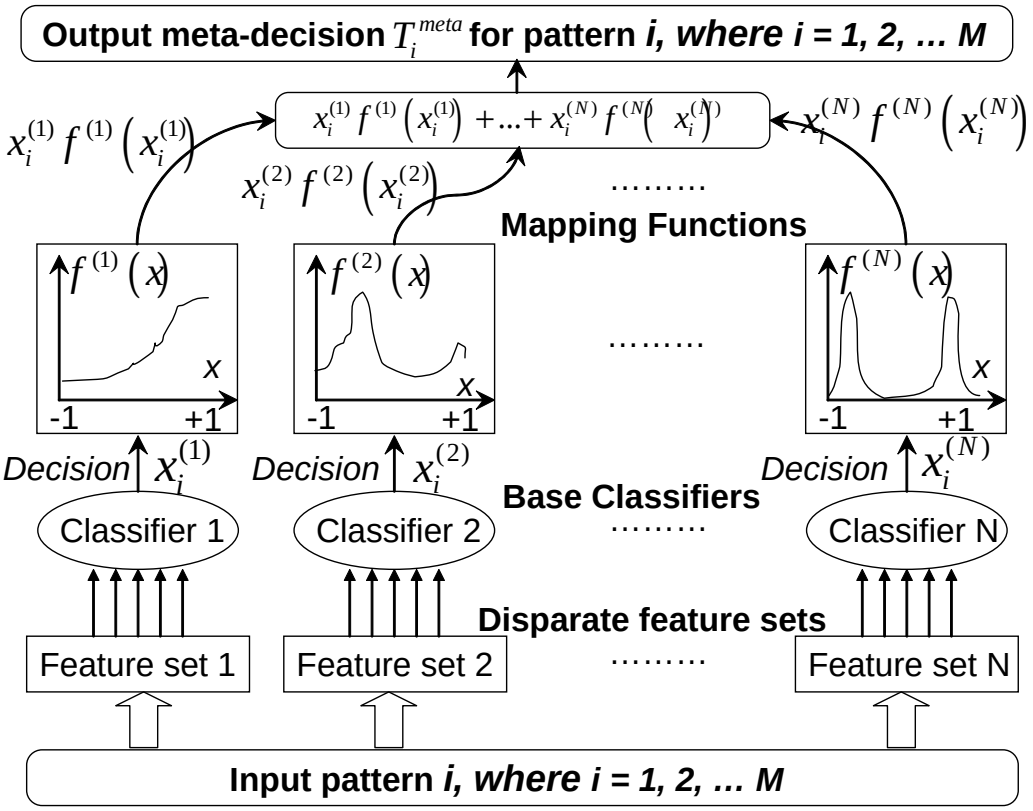


Figure 2.25: Meta-Classifier construction via statistical combination of disparate *Base Classifiers*

2.4.2.1 Meta-Classifier Construction

First we formally define the following terms as 3 major components of the *meta-classification* framework.

Definition 2.4.1. ***Meta-Classifier***: A classifier that is built and optimized via proper combinations of various *Base Classifiers* for further performance improvement and noise reduction.

Definition 2.4.2. ***Base Classifier***: A classifier whose outputs are weighed through the *Mapping Functions*, then serve as the base inputs to the *Meta-Classifier*. A *Base Classifier* can be any existing hotspot detection engine.

Definition 2.4.3. ***Mapping Function***: A weight function for each *Base Classifier* which dynamically adjusts the *Base Classifier*'s contribution to the *Meta-Classifier*. It is to be optimized properly given sufficient calibration data.

The construction of a *Meta-Classifier* is illustrated in Fig. 2.25, which is divided into 3 levels of operations. For every input data sample, certain unique feature sets are extracted then fed into corresponding *Base Classifiers*. *Base Classifiers* generate decision vectors, based on which weight vectors is generated by the *Mapping Functions*. The final output meta-decision is the weighed sum of *Base Classifiers* decisions.

Again, by examining the corner cases of the *Meta-Classifier* we notice that if the *Mapping Functions* become constant 1 for all inputs of the SVM *Base Classifier* but constant 0 for all inputs of other *Base Classifiers*, then

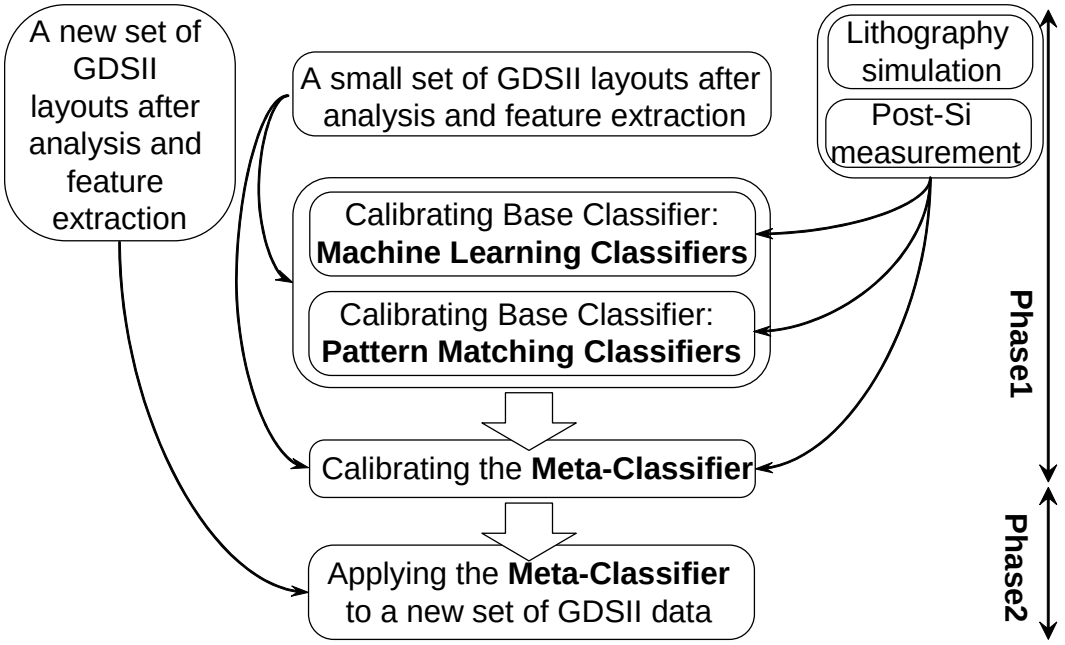


Figure 2.26: An overall CAD flow for calibrating and applying Meta-Classifiers

the *Meta-Classifier* degenerates into a SVM *Base Classifier*. This means by manipulating the *Mapping Functions* we can potentially achieve the best performance trade-off among all the options provided by the *Base Classifiers*, such as machine learning and pattern matching, etc.

2.4.2.2 Overall CAD Flow

Fig. 2.26 shows the overall CAD flow for calibrating and applying the *Meta-Classifier*. It consists of 2 phases: Phase1 is the calibration stage when the *Base Classifiers* and the *Mapping Functions* are configured and optimized using multiple data sets. This stage requires the supervision of accurate lithography simulators. Phase2 is the stage when the established *Meta-Classifier* is

applied onto new data sets. This stage operates without involving accurate but time-consuming lithography simulations.

In Phase2, the final meta-decision is calculated through the following Equation (2.25):

$$T_i^{meta} = F^{meta}\{\sum_{k=1}^{N} x_i^{(k)} \cdot f^{(k)}(x_i^{(k)})\} \tag{2.25}$$

where T_i^{meta} is the decision value of data sample i, N is the total number of *Base Classifiers*, $f^{(k)}(x)$ is the *Mapping Function* of the kth *Base Classifier*, $x_i^{(k)}$ is the output from the kth *Base Classifier* when sample i is fed as input. F^{meta} is the same as defined in Equation (2.24).

2.4.2.3 Meta-Classification Error Analysis

Given the *Meta-Classification* framework, we analyze the Mean-Square-Error of the *Meta-Classifier* introduced by the errors/noises of the *Mapping Functions* calibration stage.

Depicted in Fig. 2.27 are 2 sets of curves. The black curves are the optimal *Mapping Functions* and the intersected point *threshold** is the optimal cutoff value for $F^{th}(\cdot)$. Suppose the dotted curves are the sub-optimal *Mapping Functions* for the ith and jth *Base Classifiers*. In this case, the derived cut-off threshold becomes *threshold*+error*, therefore in the *Meta-Classifier* application stage, detection results will suffer from a MSE defined as follows,

$$MSE^{noise} = \int_x \{\sum_{k=1}^{N} f^{(k)}(x) \cdot x - \sum_{k=1}^{N} p^{(k)}(x) \cdot x\}^2 dx \tag{2.26}$$

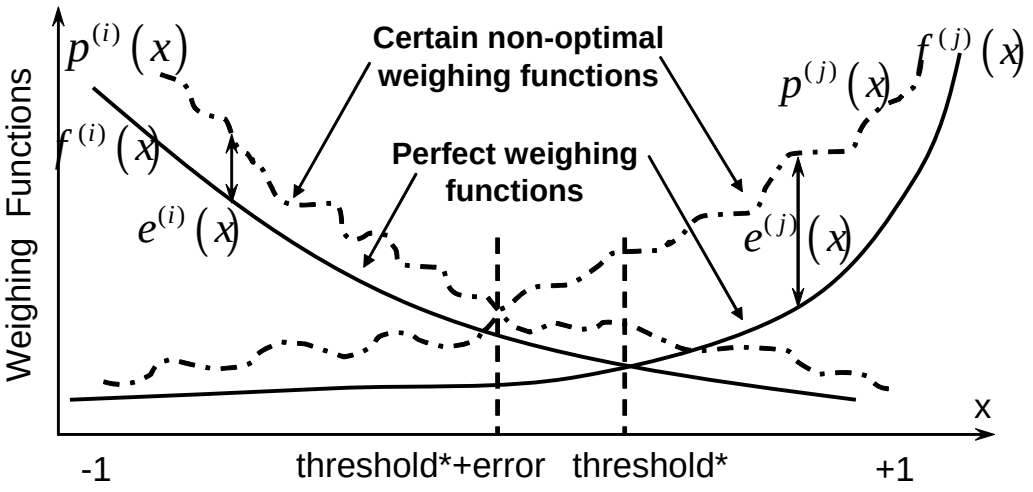


Figure 2.27: A simple illustration of the *Mapping Function* error analysis, assuming N=2

$$= \sum_{k=1}^{N} \int_x \{[f^{(k)}(x) - p^{(k)}(x)] \cdot x\}^2 dx \tag{2.27}$$

$$= \sum_{k=1}^{N} \int_x \{[\xi^{(k)}(x)] \cdot x\}^2 dx \tag{2.28}$$

$$\xi^{(k)}(x) = f^{(k)}(x) - p^{(k)}(x) \tag{2.29}$$

From the analysis above, we observe that the classification error accumulates among all *Base Classifiers* with a quadratic index on each term, should noise/error occur in the *Mapping Functions*. Therefore, it is critical to find the optimal *Mapping Functions* to ensure the *Meta-Classifier*'s noise robustness. In the following subsections, we will explore the mathematical formulation that yields the set of optimized *Mapping Functions* given certain calibration data set.

2.4.3 Mapping Function Optimization

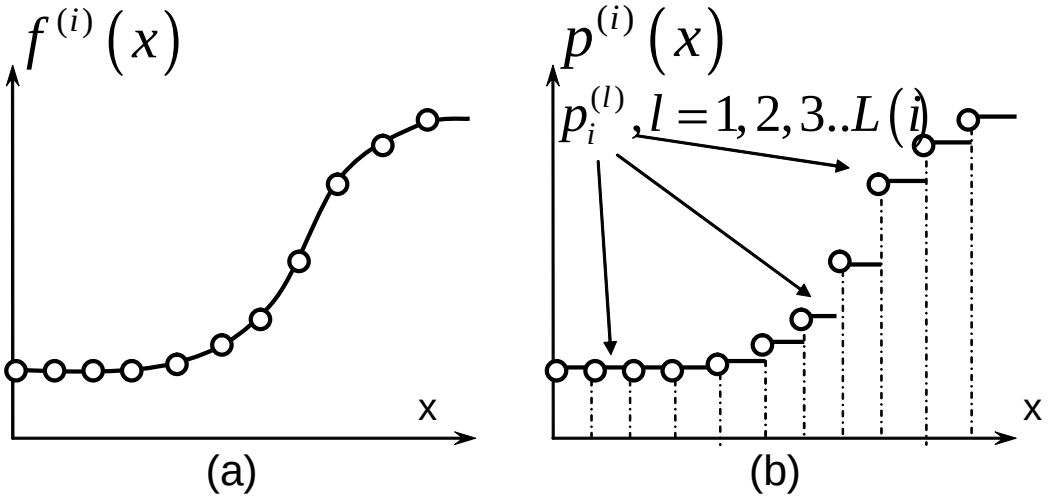


Figure 2.28: Quantization of the *Mapping Functions*

2.4.3.1 Mathematical Formulation

We first define the *Meta-Classification* Mean-Square-Error over the entire calibration data set under the supervision of accurate lithography simulations:

$$MSE^{meta} = \frac{1}{M} \cdot \sum_{i=1}^{M} \| \sum_{k=1}^{N} x_i^{(k)} \cdot f^{(k)}(x_i^{(k)}) - T_i^{meta} \|^2 \qquad (2.30)$$

where M is the total number of calibration samples. The application process is calculated as in Equation (2.25).

To minimize the Mean-Square-Error among the sample space meanwhile avoid over-fitting of the training data set, we define the performance

optimization formulation as follows,

$$To\ minimize:\ MSE^{meta}\ +\ PCost\quad w.r.t\ p^{(k)}(x_i^{(k)}) \tag{2.31}$$

$$PCost = \lambda_0 \sum_i \sum_k (x_i^{(k)} - 1)^2 \tag{2.32}$$

where λ_0 is a penalty applied to constrain the training process in order not to loss generality of the *Mapping Functions* when applied to new data sets.

As illustrated in Fig. 2.28, to reduce the solution space without obvious loss of optimality, we quantized the *Mapping Functions* $p^{(1)}(x) \sim p^{(N)}(x)$ into $L(k)$ levels, with each level being a single weight value denoted as $p_k^{(l)}$, where $l \in [1,L(k)]$.

After the *Mapping Function* quantization process, the original formulation is transformed into the following formulation:

$$To\ minimize:\ \overline{MSE}\ +\ \overline{PCost}\quad w.r.t\ p_k^{(l)} \tag{2.33}$$

$$\overline{MSE} = \frac{1}{M} \sum_{i=1}^{M} \| \sum_{k=1}^{N} p_k^{(\Theta(x_i^{(k)}))} \cdot x_i^{(k)} - T_i^{meta} \|^2 \tag{2.34}$$

$$\overline{PCost} = \lambda_0 \sum_{k=1}^{N} \sum_{l=1}^{L^{(k)}} (p_k^{(l)} - 1)^2 \tag{2.35}$$

where $p_k^{(l)}$'s are the optimization variables.

Therefore we can write a quadratic programming problem formulation:

$$f(x) = \frac{1}{2} X^T Q X + c^T X \tag{2.36}$$

$$X \geq lb \tag{2.37}$$

$$lb = [0\ 0\ 0\ 0\ 0\ ...\ 0]^T \in \Re^{L^{total}\times 1} \tag{2.38}$$

where X is the optimization variable vector defined as follows,

$$X = [p_1^{(1)}\ ...\ p_1^{(L^{(1)})}\ ...\ p_k^{(l)}\ ...\ p_N^{(L^{(N)})}]^T \in \Re^{L^{total}\times 1} \tag{2.39}$$

where L^{total} is the total number of $p_k^{(l)}$'s:

$$L^{total} = \sum_{k=1}^{N} L^{(k)} \tag{2.40}$$

where matrix Q is defined as follows, $Q\ =$

$$\begin{pmatrix} \beta_1^{(1)}(i) & \gamma_{1,1}^{(1,2)}(i) & . & \gamma_{1,k}^{(1,l)}(i) & \gamma_{1,N}^{(1,L^{(N)})}(i) \\ \gamma_{1,1}^{(2,1)}(i) & \ddots & . & . & . \\ . & . & \beta_1^{(L^{(1)})}(i) & . & . \\ \gamma_{k,1}^{(l,1)}(i) & . & . & \beta_k^{(l)}(i) & . \\ \gamma_{N,1}^{(L^{(N)},1)}(i) & . & . & . & \beta_N^{(L^{(N)})}(i) \end{pmatrix} \tag{2.41}$$

and vector c is defined as the linear term coefficients vector from the quadratic formulation objective:

$$c = [\omega_1^1(i) \cdots \omega_1^{L^{(1)}}(i) \cdots \omega_k^{(l)}(i) \cdots \omega_N^{L^{(N)}}(i)\ ]^T \tag{2.42}$$

where the related terms are defined as follows, and $\alpha_k^{(l)}(i)$ is an intermediate term to link the *Base Classifier*'s prediction values with $p_k^{(l)}$, i.e., $\alpha_k^{(l)}(i)$ is the $x_i^{(k)}$ value that falls into level l of the quantized mapping function relating to the *Base Classifier* k. Given certain i and k, if there is no output values corresponding to level l, then $\alpha_k^{(l)}(i)$ is set to 0.

$$\beta_k^{(l)}(i) = \frac{2}{M}\sum_{i=1}^{M} [\alpha_k^{(l)}(i)]^2 + 2\lambda_0 \tag{2.43}$$

Algorithm 10 ***Meta-Classifier*-Calibration**

Require: data sample vectors and over-fit penalty λ_0
  Initialize Q, c, $\beta_k^{(l)}(i)$, $\gamma_k^{(l)}(i)$, $\omega_k^{(l)}(i)$
  Build Hierarchical *MLK-ANN*[42]
  Build Hierarchical *MLK-SVM*[42]
  Build Pattern Matchers
  for All input data samples **do**
    Generate the *Base Classifiers*
    Update Q, c, $\beta_k^{(l)}(i)$, $\gamma_k^{(l)}(i)$, $\omega_k^{(l)}(i)$
  end for
  Formulate Quadratic Programming Problem
  if Q not positive definite **then**
    Request for more calibration data
  end if
  Solve the Quadratic Programming Problem
  Optimize the detection threshold function $F^{meta}(\cdot)$
  return *Mapping Functions* levels $p_k^{(l)}$ and $F^{meta}(\cdot)$

$$\gamma_{m,k}^{(n,l)}(i) = \frac{2}{M} \sum_{i=1}^{M} \alpha_m^{(n)}(i) \cdot \alpha_k^{(l)}(i) \tag{2.44}$$

$$\omega_k^{(l)}(i) = -\frac{2}{M} \sum_{i=1}^{M} T_i^{meta} \cdot \alpha_k^{(l)}(i) - 2\lambda_0 \tag{2.45}$$

For a quick reference, Table 2.9 details the variables and terms used in our quadratic programming formulation.

Table 2.9: Variables and terms in the QP formulation

Terms	Descriptions
N	Number of the *Base Classifiers*
M	Number of meta-machine calibration patterns
i	Index of the input pattern samples
k	Index of the base classifiers
$x_i^{(k)}$	Prediction result from the base classifier k, given the input data sample i
$f^{(k)}(x_i^{(k)})$	The value of perfect mapping func $f^{(k)}(\cdot)$ at $x_i^{(k)}$
$p^{(k)}(x_i^{(k)})$	The value of certain mapping func $p^{(k)}(\cdot)$ at $x_i^{(k)}$
$L^{(k)}$	Total quantization levels of base classifier k
l	Index of the quantization levels
$p_k^{(l)}$	Quantized weight value from $f^{(k)}(\cdot)$ for level l of the base classifier k, $p_k^l \in [1,\ L^{(k)}]$
$\Theta(\cdot)$	quantization mapping func.: $x_i^{(k)} \rightarrow$ level index l
$\alpha_k^{(l)}(i)$	The *Base Classifier* output to which $p_k^{(l)}$ is applied, i.e., the $x_i^{(k)}$ relating to level l of base classifier k given pattern i. (0 if NULL)
L^{total}	Total number of independent $p_k^{(l)}$
Q	A definite positive matrix $\in \Re^{L^{total} \times L^{total}}$
c	A vector $\in \Re^{L^{total} \times 1}$
X	Variable vector for the quadratic programming $X = [p_1^{(1)}\ ...p_k^{(l)}\ ...p_N^{(L^{(N)})}]^T \in \Re^{L^{total} \times 1}$
T_i^{meta}	Meta machine output given input sample i
λ_0	Coefficient to avoid over-fitting and instability

Algorithm 11 ***Meta-Classier*-Application**

Require: data sample vectors
  Load *Mapping Functions* $p_k^{(l)}$ and $F^{meta}(\cdot)$
  Load all *Base Classifiers*
  Generate vector $\overrightarrow{x_i}$ from *Base Classifiers* outputs
  for Each data vector $\overrightarrow{x_i}$ **do**
    Calculate T_i^{meta}=$F^{meta}(\sum_{k=1}^{N} P_k^{(\aleph(x_i^{(k)}))} \cdot x_i^{(k)})$
  end for
  return Meta-decision $\{T_i^{meta}\} \subset \{\{-1\},\{+1\}\}$

2.4.3.2 Complexity Analysis

In examining the above formulation, we can draw the following conclusions about its solution optimality and time complexity:

Remark 2.4.1. Q matrix is positive definite (Q has no negative eigenvalues) under certain conditions.

Proof 2.4.1. *For notation simplicity, we assume a vector* $\vec{X} \in \Re^{L^{total}\times 1}$ *and* $X \geq \vec{0}$. *Let* ρ_i *be the coefficient of* χ_i, *where* χ_i *is the* i*th element of* $\vec{X}$. *Let* L^{total} *be the total number of quantization levels among all* Base Classifiers. *Therefore we have the following:*

$$\vec{X}^T Q \vec{X} = \frac{1}{M} \sum^{M} \sum_{j=i+1}^{L^{total}} \sum_{i=1}^{L^{total}} (\rho_i \chi_i + \rho_j \chi_j)^2 + \Delta \tag{2.46}$$

where

$$\Delta = -\frac{L^{total}-2}{M} \sum^{M} \sum_{i}^{L^{total}} \rho_i^2 \chi_i^2 + \lambda_0 \sum_{i}^{L^{total}} \chi_i^2 \tag{2.47}$$

Under the condition when λ_0 *is larger than* $-\frac{L^{total}-2}{M} \sum^{M}$, Δ *is larger than 0, therefore* $\vec{X}^T Q \vec{X}$ *is always positive. Thus Q is a positive definite matrix and*

has no negative eigenvalues under the specified condition. Numerical simulations further validate this proof by showing all positive eigenvalues for Q.

Remark 2.4.2. The formulated Quadratic Programming problem can be solved in polynomial time complexity under certain conditions.

Proof 2.4.2. *Since matrix Q is conditioned to be positive-definite with non-negative eigenvalues, the Quadratic Programming problem can be solved by the* ellipsoid method*[72] in polynomial time.*

Remark 2.4.3. If the formulated Quadratic Programming problem has a local minima, then this local minima is also the global minima.

Proof 2.4.3. *Q is conditioned to be a positive definite matrix and $f(\cdot)$ is a convex function. The quadratic program has a global minimizer if there exists some feasible vector X^n satisfying the constraints and if $f(\cdot)$ is bounded below on the feasible region ($X^n \in \Re_+^n$). Therefore in search of a local minima, if found, will guarantee the optimal global minima.*

Based on the above theorems and discussions, we solve the proposed quadratic programming problem with the optimal *Mapping Functions* during the calibration/training stage. Then a heuristic approach is employed to search for the optimal $F^{meta}(\cdot)$ function given a set of *Mapping Functions*. In Algorithm 10 and Algorithm 11, we show the detailed operations and procedures performed for the calibration and application of the *Meta-Classifier*.

2.4.4 Base Classifier Construction

We elaborate the *Base Classifiers* we build using machine learning techniques (Artificial Neural Network and Support Vector Machine) and pattern matching techniques.

2.4.4.1 Artificial Neural Network Classifiers

The ANN *Base Classifier* is build via fine tuning[42]. Here we give the formulation as follows, further details please refer to Algorithm 4.

$$objective : minimize\{\sum_{p=1}^{N} E^p\} \quad w.r.t \quad \omega_{ij}, \omega_{jk} \tag{2.48}$$

$$E^p = \frac{1}{2}[out_p - y_p]^2 \tag{2.49}$$

$$out_p = f_{out}\{\sum_j \omega_{jk} \cdot f_{hid}(\sum_i V_p^i \cdot \omega_{ij})\} \tag{2.50}$$

$$\frac{\partial E^p}{\partial \omega_{jk}} = (out_p - y_p) \cdot f_{hid}\{\sum_i V_p^i \cdot \omega_{ij}\} \tag{2.51}$$

$$\frac{\partial E^p}{\partial \omega_{ij}} = (out_p - y_p) \cdot \omega_{jk} \cdot V_p^i \cdot (1 + out_{hid}^j)(1 - out_{hid}^j) \tag{2.52}$$

$$f_{hid} = \frac{2}{(1 + e^{-2x})} - 1, \qquad f_{in} = f_{out} = x \tag{2.53}$$

$$sign\_func(x) = \begin{cases} -1 & \text{x} < 0 \\ 0 & \text{x} = 0 \\ +1 & \text{x} > 0 \end{cases} \tag{2.54}$$

$$Est_{\tilde{p}} = F^{ann}\{f_{out}[\sum_j \omega_{jk} \cdot f_{hid}(\sum_i V_{\tilde{p}}^i \cdot \omega_{ij})]\} \tag{2.55}$$

Here the ANN classifies data by predicting a value for each sample vector V_p based on an established set of weights and biases assigned to certain neural network structure. The ANN models are customized with single hidden layer of neurons, with transfer functions denoted as f_{hid}. Inputs V_p to the ANN kernels are the extracted feature vector samples labeled with values (y_p) indicating hotspot or nonhotspot patterns (these values can be continuous for variability prediction). We use p to represent feature vector index with $p = 1$ to N, V_p^i denotes the ith element of vector V_p, $i = 1$ to M, where M is the total number of features for each sample vector. We use f_{in} and f_{out} to represent input and output layer transfer functions, and index i, j, k to indicate neuron indices in the input, hidden and output layer respectively. Once the ANN *Base Classifier* is fully calibrated, we can apply it to identify hotspots according to Algorithm 5 without using costly lithography simulations.

2.4.4.2 Support Vector Machine Classifiers

Inside the meta machine block, we employ a C-class Support Vector Machine (SVM) classifier fine-tuned based on[42]. We present the problem formulation again. Please refer to Algorithm 6 for more details.

$$objective : minimize\{f(\alpha) = \frac{1}{2}\alpha^T M\alpha - e^T\alpha\} \quad w.r.t \quad \alpha \tag{2.56}$$

$$subject \quad to : 0 \leq \alpha_i \leq C, i = 1, ..., N, \tag{2.57}$$

$$y^T \cdot \alpha = 0 \tag{2.58}$$

$$K(V_i, V_j) = exp\{\gamma \cdot \|V_i - V_j\|^2\} \tag{2.59}$$

$$slope\_func(x) = \begin{cases} 0 & \text{x} \leq 0 \\ x & 0 < \text{x} < \text{C} \\ C & \text{x} \geq \text{C} \end{cases} \tag{2.60}$$

$$Est_{\tilde{p}} = F^{svm}\{\sum_i \alpha_i y_i K(V_{\tilde{p}}, V_i) + bias\} \tag{2.61}$$

Given V_i, i=1 to M sample vectors with N number of features, with label y_i (either +1 or -1 for 2-class SVM). e is a vector of all 1's. C is a pre-set upper bound to constrain feasible regions for hotspot detection under real manufacturing conditions. M is N by N positive semi-definite matrix defined as $M_{ij} = y_i y_j K(V_i, V_j)$, where $K(V_i, V_j)$ is defined in Eqn.(2.59) as the kernel function. α is the N element weight vector for V_p's. Note α is generally sparse and the non-zero weights correspond to the final support vectors. Due to the fact that M is usually dense and large, decomposition methods are usually used to solve the formulation iteratively rather than directly dealing with the quadratic Eqn.(2.56).

The configuration of SVM *Base Classifiers* are achieved through performing Algorithm 6, which intakes data set V_p's and returns the supporting vectors and corresponding weight coefficients. There are 3 major steps involved: First, data set normalization for detection robustness; Second, high order working set selection for enhanced detection accuracy particularly for our special hotspot detection requirements; Third, update weight and gradient vectors. The last 2 steps are carried out in an iterative manner until certain error target is met. For implementation details regarding the higher order working set selection please refer to [49]. When the SVM model is fully

trained and configured, we can apply it to evaluate a new design pattern using Algorithm 7, which requires no more accurate lithography simulations.

2.4.4.3 Pattern Matching Classifiers

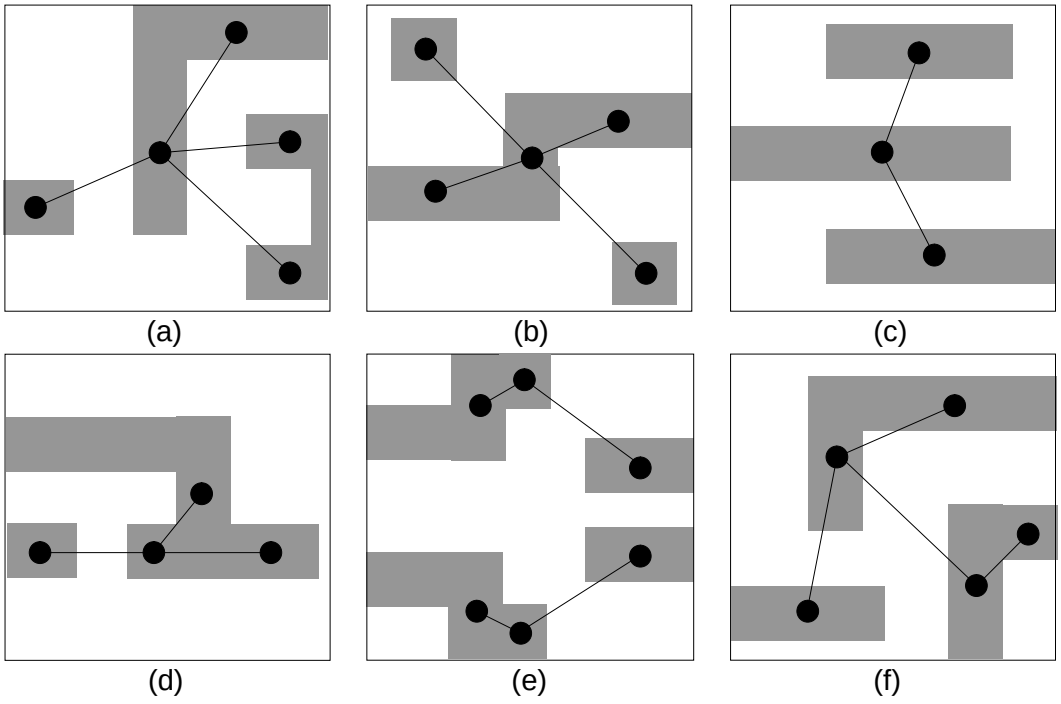


Figure 2.29: Example hotspot patterns covered by PM *Base Classifiers*

We explored the current state-of-the-art methods [66, 135, 138] and come up with 4 Pattern Matching *Base Classifiers* to cover various disparate classes of lithography hotspots, relating to special line-end, corner/jogs and contact patterns, etc. Some hotspot example patterns are illustrated in Fig. 2.29. In particular, we have fine-tuned the pattern matchers to have broader pattern coverage rather than performing exact matching. As a result, the established

pattern matchers demonstrate very good hotspot accuracies onto new data sets. Obvious the penalty of such fine-tuning is the resulting high false-alarms. However, as we will see later in Subsection 2.4.5, the *Meta-Classifier* performs well in suppressing the false-alarms of such a PM *Base Classifier*.

2.4.5 Simulation and Testing

2.4.5.1 Testing Benchmarks

Table 2.10: Circuit benchmarks for testing *EPIC*

Benchmarks	CK1	CK2	CK3
Layout Size um^2	85×85	100×120	550×600
Fragment number	58K	94.5K	2.5M
Class0[a] Hotspots number	9	21	122
Class1[b] Hotspots number	61	134	2.8K

[a] Class0 hotspots: EPE $\geq 6nm$ for $32nm$ process.
[b] Class1 hotspots: $4.5nm \leq$ EPE $< 6nm$ for $32nm$ process.

EPIC employs industry-strength benchmarks targeted for 32nm process technologies with industry-strength Resolution Enhancement Techniques and manufacturing conditions.

To fully evaluate *EPIC*, we employed several training data sets and 3 testing circuit benchmarks in 32nm. We defined 2 categories of lithography hotspots based on different ranges of EPE values. As shown in Table 2.10, C0 refers to hotspots with more than 6nm of EPE, C1 refers to hotspots with EPE in between of 4.5nm and 6nm.

2.4.5.2 Experimental Setups

EPIC incorporates the successively-refined machine learning methods proposed by [42] and several state-of-the-art pattern matching techniques based on [66, 135, 138].

Simulation experiments are implemented in C/C++ and performed on 3.2GHz quad-core Linux workstations. Accurate lithographic simulations are carried out under industry-strength RET and manufacturing conditions. There is no lithography simulations involved in the testing flow once the *Meta-Classifier* and *Mapping Functions* are calibrated.

2.4.5.3 Result Analysis and Comparison

To analyze *EPIC*, we report the simulation results and compare it with other types of hotspot detection methods in terms of accuracy, false-alarm and performance trade-off capability.

In Fig. 2.30 we fine-tune the decision threshold functions of *EPIC*, ANN classifier and SVM classifier to plot their respective performance trade-off regions. We also plot the performance region of the employed pattern matchers, which include 4 major classes of hotspot patterns. As we enrich the pattern library with up to hundreds of specific patterns and structures using the training data, a good detection coverage over the testing data (with some previously unseen hotspot patterns) becomes a trade-off between enhancing detection generality and suppressing false-alarms. There are two observations: first, *EPIC* achieves the best performance trade-off reported so far. In par-

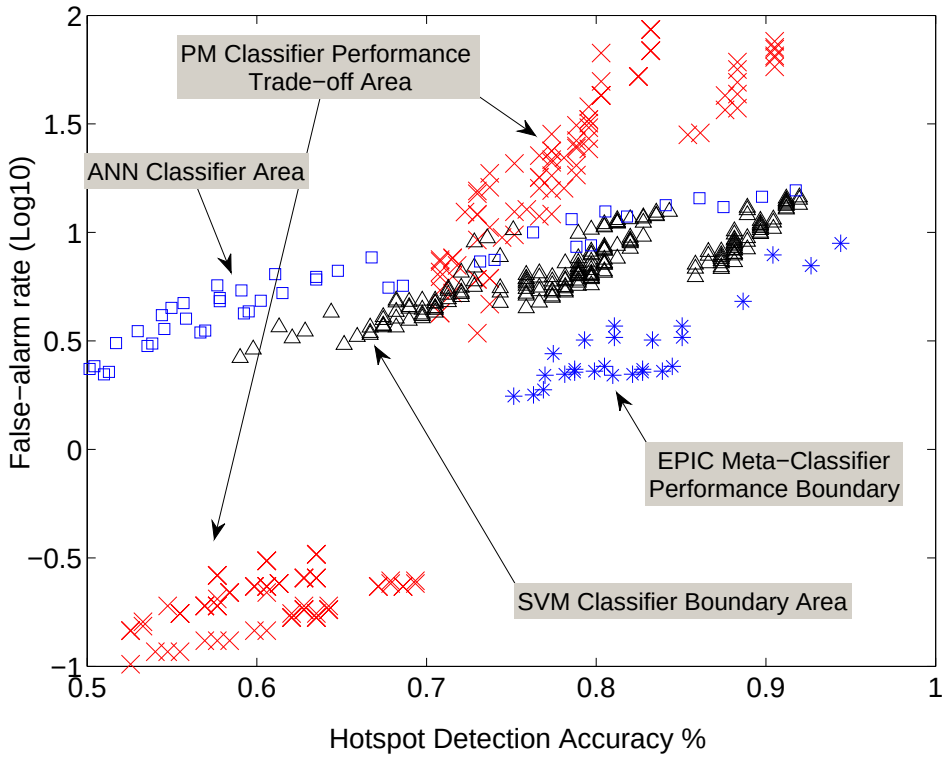


Figure 2.30: Trade-off capabilities between hotspot accuracy and false-alarms using various methods on C0 hotspots at 32nm

ticular, for the region of above 70% accuracy, *EPIC* shows higher detection accuracy than other methods with the same false-alarm rate, meanwhile lower false-alarms than other methods that have the same accuracy rate. Second, pattern matching methods are not good at detecting new types of hotspots without obvious penalty in false-alarm rate. In this sense, machine learning can make pattern matching more robust to predict new/unknown hotspots, especially when pattern enumeration becomes too costly.

Based on Fig. 2.30, we calculate the following for each method:

$$\Psi = \alpha \cdot Accuracy^{hotspot} + \beta \cdot False\_alarm^{hotspot} \tag{2.62}$$

where α and β are user defined parameters and set to 1:(-0.2). In Table 2.11 and Table 2.12, we report the detection result of each method corresponding to the peak of their respective Ψ function. Here we have the following observations. First, EPIC reaches very high performance for both C0 and C1 hotspots. Second, EPIC improves ANN and SVM by 3.5-7% in accuracy and up to 50% in false-alarm reduction, meanwhile it outperforms PM by 4-12% in accuracy and above 80% in false-alarm reduction. This demonstrates very promising potential and effectiveness of the *Meta-Classification* flow with respect to *Mapping Function* optimizations. Third, EPIC runs at the speed of around 45min per mm^2 design on a 3.2GHz quad-core workstation, which is typically hundreds of time faster than accurate lithography simulator.

Moreover, *EPIC*'s unified formulation covers the hybrid detection flow proposed in [131] as a special corner case, i.e., when *Mapping Function* $f^{MLK1}(+1) =$ 0.5 and 0 elsewhere, $f^{MLK2}(+1) = 0.5$ and 0 elsewhere, $f^{PM}(+1) = 1.0$ and 0 elsewhere, $\theta = 1.0$, then *EPIC*'s formulation Equation (2.25) will be equivalent to the hybrid flow in [131]. Here *EPIC*'s advantage lies in the automated optimization techniques, thus it can easily reach an optimized solution.

As the simulation results show, *EPIC* achieves so far the most promising performance trade-off and flexibility between detection accuracy and false-alarm suppression.

Table 2.11: Performance report of *EPIC*

Circuits	Class	Perf.	ANN	SVM	PM	*EPIC*
CK1	C0	Hit	6	7	7	9
		Extra	79	41	433	48
	C1	Hit	52	54	53	57
		Extra	0.55K	0.33K	1.5K	0.3K
CK2	C0	Hit	17	17	16	19
		Extra	0.2K	0.11K	1.2K	0.1K
	C1	Hit	119	120	120	125
		Extra	1.2K	0.75K	3.4K	0.65K
CK3	C0	Hit	109	108	99	112
		Extra	1.2K	0.6K	6.7K	0.65K
	C1	Hit	2.45K	2.5K	2.5K	2.6K
		Extra	24K	16K	73K	13.5K

Table 2.12: Comparison between *EPIC* and previous works

Hotspot	C0			C1		
Avg. Perf.	Hit	Extra	Time	Hit	Extra	Time
EPIC	92%	5X	0.72	94%	4.8X	0.72
ANN[41, 42]	89%	10X	0.3	88%	8.8X	0.3
SVM[41, 42]	86%	5X	0.35	89%	5.5X	0.35
PM[66, 83, 135, 138]	82%	56X	0.2	90%	25X	0.2

Run-time in $hour/mm^2$ on 3.2GHz quad-core Linux.

2.4.6 Summary

In this section, we presented EPIC: a generic and unified methodology to seamlessly combine the advantages of various lithography hotspot detection techniques for the ultimate performance enhancement. We developed several machine learning and pattern matching detection engines as base classifiers and unified them under an optimized meta-classification flow using convex programming techniques. We evaluated EPIC with various industry bench-

marks under advanced manufacturing conditions to demonstrate its capability to select the desirable features of all base classifiers meanwhile greatly suppress the detection noise. The demonstrated potentials of EPIC makes it very suitable for ultra high performance physical verification and for guiding DFM in the physical design stages.

2.5 Generic Lithography-Friendly Detailed Routing with Post-RET Data Learning and Hotspot Prediction

As a consequence of the shrinking technology process and the increasing chip complexity, the design and manufacturing cycles start to see more and more interactions for modern VLSI ICs. As lithography-induced yield is a critical factor to optimize for volume manufacturing of nanometer ICs, various resolution enhancement techniques (RET), such as optical proximity correction (OPC), off-axis illumination (OAI), phase shift mask (PSM), source-mask optimization (SMO), and double patterning technology (DPT) are employed to enhance layout printability and yield.

However, RET during mask synthesis alone is not enough due to widening manufacturing gaps [103], which require increasing cooperation of physical design methods to generate lithography-friendly layouts to start with. Several works have been proposed to incorporate accurate lithographic models or predictive models into physical design stages, in particular the routing stages, to ensure layout printability. In [89], post-routing ripup-&-reroute was proposed to remove lithography hotspots, guided by fast lithography simulations. In [27], an OPC-cost aware router was proposed utilizing post-layout OPC models characterized by quasi-inverse lithography techniques. OPC-aware maze routing methods are also proposed based on multi-constrained shortest path optimization with sub-gradient method [60] and optical proximity error (OPE) metrics [130]. In [33], a litho-friendly detailed router was proposed based on weak grid types of predictive OPC metrics that are updated on a per-grid ba-

sis. However, these existing studies all suffer from one or more of the following issues: (1) huge run-time due to accurate but slow lithographic simulations; (2) over simplified predictive models severely limit the solution space; (3) particularly designed to work only for certain classes of RETs (e.g., OPC, etc.) under certain data fitting assumptions, but not generic enough to handle new types of RETs (e.g., Litho-Etch-Litho-Etch Double Patterning, Self-Aligned Double Patterning, Sub-Resolution Assist Features Insertion) and evolving manufacturing conditions.

To address these concerns, modern graph theory and data mining/learning methods have recently been adopted to build reliable and high performance lithography hotspot detection engines. [66] proposed a graph pattern based hotspot filtering method to reduce the hotspot candidate searching space without compromising the overall detection quality. Concept of range pattern is later introduced in [138] to accurately and compactly represent process hotspots. In [137], ripup-&-reroute techniques are proposed for hotspot removal, utilizing a pre-defined pattern matching library. Although generally fast, issues with graph/pattern matching techniques lie in detection coverage and scalability: (1) hotspot patterns are very difficult to enumerate - too much patterns may constrain the solution space, while too few patterns suffer from high detection false-alarms; (2) high false-alarms in the physical design stages introduce heavy workload for post synthesis correction; (3) pattern definition is highly dependent on design rules and process technology, therefore increases the development burden as technology evolves.

With data mining techniques, further improvements were later made in [47], where a support vector machine (SVM)-based hotspot detection method is utilized through performing 2D distance transform and histogram extraction. Also in [133], SVM is employed for hotspot detection through classification of layout density metrics. However, the issues with the above approaches lie in run time and detection coverage, since 2D transforms and density extractions can be expensive to perform, while detection windows for the layout images are hard to anchor for full chip level detections. In [43], critical hotspot signature is proposed and extracted through certain special edge-based metrics. Although such edge-based extractions operate much faster compared with [47, 133], its chip-level prediction still faces similar issues, such as scanning window coverage, etc. To further improve runtime and detection coverage, a hierarchically refined machine learning framework is proposed in [42] for fast speed high performance hotspot detection using both Artificial Neural Network (ANN) and SVM classifications.

Yet there has been no work so far to apply such data learning hotspot detection framework directly to physical design in a correct-by-construction manner. This is mainly because hotspot detection requires post layouts as inputs, thus the lithography cost cannot be updated in time to guide the physical design. To address the aforementioned limitations, we propose *AENEID*, a lithography-friendly detailed router that is seamlessly integrated with modern data learning techniques and predictive models for litho-aware path prediction.

The rest of the section is organized as follows, in Subsection 2.5.1, we

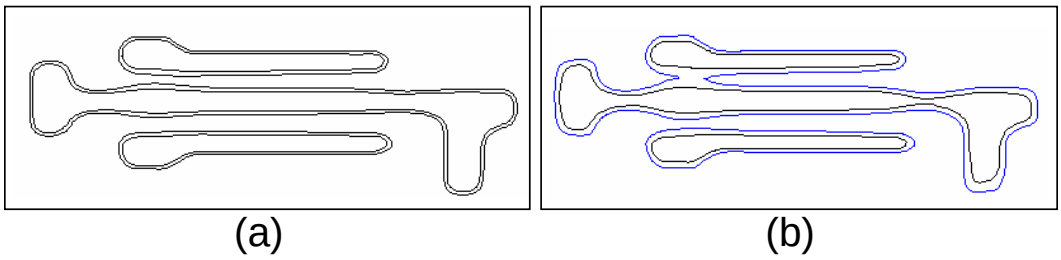


Figure 2.31: A case of RET dependent layout printability

elaborate the motivation of this work and summarize our main contributions. In Subsection 2.5.2 and 2.5.3 we explain the formulation and overall flow of *AENEID*. In Subsection 2.5.4, we describe the key novel techniques employed for the hotspot detection and routing path prediction procedures. In Subsection 2.5.5 we evaluate *AENEID* with various industry strength benchmarks under industry strength RET, we also present and discuss the experimental results compared with some existing study. We conclude the section with a brief summary in Subsection 2.5.6.

2.5.1 Motivation and New Contributions

In this subsection, we explain the post-RET hotspot detection dilemma in the detailed routing stage and introduce two most critical considerations in *AENEID*: (1) fast and accurate lithographic hotspot characterization that drives the litho-friendly router; (2) look-ahead routing path prediction that adjusts the litho-cost assigned to the router for enhanced yield and routability.

With the rapid advancement of modern lithography technology and RET techniques, real process hotspots are getting less as process matures

and RET improves, however real/residual lithography hotspots are becoming more critical to the nanometer IC design. In the mask synthesis stage, process hotspots are highly dependent on the RETs employed: an effective RET can print a layout pattern reliably (Fig. 2.31(a)), whereas a poorly setup RET may generate many false-alarm "hotspots" (Fig. 2.31(b)) that will not be present under real industry-strength manufacturing condition[42]. With such considerations, data learning models [42, 43, 47, 133] become especially suitable for guiding detailed routers due to their satisfactory performance. However, they must be properly sped up before incorporating into inner design loops, due to the strict run-time requirement in the routing stage.

In addition to strict run-time requirement, another challenge lies in the proper modification of data learning kernels to comply with the incremental mechanism in the detailed routing stage. With the example in Fig. 2.32(a)-(b), we show a hotspot detection dilemma in the detailed routing stage.

In Fig. 2.32(a), the shadowed regions are metal blockages; Pin1-Pin2, Pin3-Pin4 are 2 nets yet to be routed in the detailed routing stage, which means at the current step the bottom-right corner is a blank region. On one hand, the detailed routing paths from Pin1 to Pin2 and from Pin3 to Pin4 are to be optimized depending on the cost updates to be provided by a hotspot detection engine; on the other hand, a hotspot detection engine must first have a routing path in order to provide the routing cost updates. In other words, the lithography cost cannot be updated in time to guide routing in a correct-by-construction manner. Consequently, we are left with a un-characterized

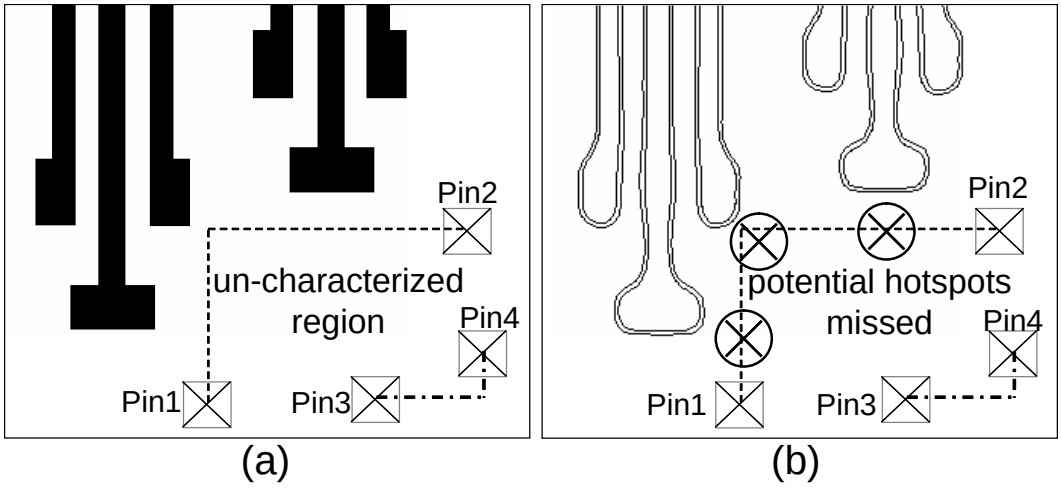


Figure 2.32: The lithography hotspot detection dilemma in the detailed routing stage

region shown in Fig. 2.32(a). Within this region, detailed routing might be performed in a lithography unaware manner which will potentially result in large amount of lithography hotspots (Fig. 2.32(b)).

To eliminate un-characterized regions, we propose for the first time a novel set of predictive formulae on top of existing hotspot detection kernels, to predict the routing path with the least *expected* lithographic cost. Such a set of formulae is developed *a priori* through accurate lithographic simulations of various layout samples at a one time cost. Once completed, it can be rapidly applied to compensate existing hotspot detectors in the detailed routing stage. More details will be explained in Subsection 2.5.4.

With these motivations and considerations, we propose *AENEID*, the first correct-by-construction detailed router formulation incorporating advanced

data learning techniques and routing path predictive formulae for lithographic yield improvement. We summarize the main contributions of *AENEID*:

- We propose a correct-by-construction detailed routing flow that is *generic* and *adaptive* to any existing RETs without any lithographic simulations involved during the routing stage.
- We employ modern data learning techniques for fast and accurate lithography hotspot detection.
- We develop lithography-friendly routing path prediction model to resolve the hotspot detection dilemma in the detailed routing stage.
- We integrate the hotspot detection and routing path prediction techniques into a scalable and high performance litho-friendly detailed router and achieve very promising results.

2.5.2 Problem Formulation

The objective in our detailed routing is to minimize total wirelength (number of grids in set P of total routing paths) with the constraint to keep the lithography cost $litho(e)$ on each routing grid e under a given threshold L. Therefore we can formulate our lithograph-friendly detailed routing problem:

$$\begin{aligned} \min_{P}: \quad & \textstyle\sum_{e \in P} 1 \\ s.t: \quad & litho(e) \leq L \quad \forall e \in P \end{aligned} \tag{2.63}$$

If we treat the costs for all the grids as a weight-vector, this problem can be viewed as a multi-constraint shortest path (MCSP) problem [45] which is proven to be NP-hard. Lagrangian relaxation can be used to solve MCSP by relaxing the constraints into the objective function by introducing Lagrangian multiplier λ_e as the weights on the constraints. Then we can relax the original formulation as [33, 60]:

$$\begin{aligned} &\min_P \left\{ \sum_{e \in P} 1 : litho(e) \leq L, \forall e \in P \right\} \\ &\geq \max_{\lambda} \min_P \sum_{e \in P} 1 + \lambda_e (litho(e) - L) : \lambda_e \geq 0 \end{aligned} \tag{2.64}$$

Equation(2.64) shows that the optimal solution of Equation(2.63) can be obtained by maximizing the lower bound of the following Lagrangian sub-problem:

$$\begin{aligned} \min_P : \quad & \textstyle\sum_{e \in P} 1 + \lambda_e (litho(e) - L) \\ s.t. : \quad & \lambda_e \geq 0 \qquad \forall e \in P \end{aligned} \tag{2.65}$$

After assigning $1+\lambda_e litho(e)$ as the weight of each grid e, Equation(2.65) can be solved by min-cost path algorithm. Then we can iteratively solve Equation(2.65) and adjust the Lagrangian multiplier to obtain the maximum lower bound for the relaxed problem in Equation(2.64). Based on the formulation here, we explain our detailed router in Subsection 2.5.3.

2.5.3 Overall CAD Flow

Since maximizing Equation(2.65) is a convex programming problem, we can apply subgradient method to solve it. As shown in Fig. 2.33, the key steps of *AENEID* are as follows.

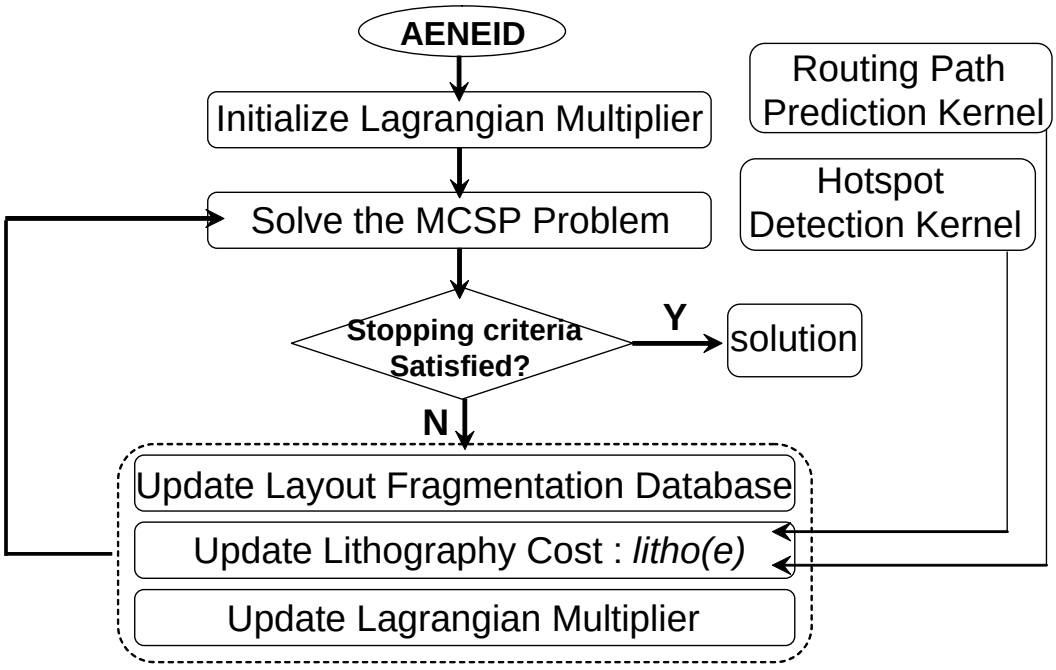


Figure 2.33: *AENEID* detailed routing flow chart

- **Step 1**: Initialize Lagrangian multipliers λ_e with small non-zero values for faster convergence.
- **Step 2**: For each net, solve Equation(2.65) by finding its min-cost path. A*-tree technique is used here to prevent unnecessary path finding.
- **Step 3**: update λ_e by $\max(0, \lambda_e + \theta \times litho(e))$ according to the result

from **Step 2**. Here the update of $litho(e)$ consisting of two parts, whose values are calculated by 2 pre-established kernels.

- **Step 4**: repeat from **Step 2** until the maximum iteration number is reached. Finally we can obtain a convergent solution.

In Fig. 2.33, **Step 3** is the most critical step for *AENEID*, since it is in charge of litho-cost updates for every loop of the detailed routing. Inside the *Lithography Cost Update* procedure of **Step 3** are the two most important contributions of this work, namely the *Hotspot Detection (HD)* technique and the *Routing Path Prediction (RPP)* technique, which we will elaborate in the following subsection.

2.5.4 Data Learning and Hotspot Prediction

In the detailed routing flow in Subsection 2.5.3 we represent the layout contents (polygons) with fragmentation-structured database meanwhile maintain a litho-cost map for the entire set of routing grids according to the current routing density. The lithography cost $litho(e)$ is then iteratively updated for each grid e. In *AENEID*, $litho(e)$ is calculated in two parts in Equation(2.66):

$$litho(e) = litho(e)^{HD} + litho(e)^{RPP} \tag{2.66}$$

where $litho(e)^{HD}$ is calculated using the *Hotspot Detection* technique and $litho(e)^{RPP}$ is calculated with the *Routing Path Prediction* technique. Depending on whether the neighborhood has been populated with wire segments

or not, $litho(e)^{HD}$ and $litho(e)^{RPP}$ combine hotspot litho-cost of the current step layout and of the future routing steps together to provide enhanced lithography-friendliness and improved routability. Unlike the existing works such as [33] that only considers $litho(e)^{HD}$ term when updating litho-cost metrics, *AENEID* benefits greatly from such a combination of litho-costs that we propose, as we will show later in Subsection 2.5.5.

2.5.4.1 Hotspot Detection Technique

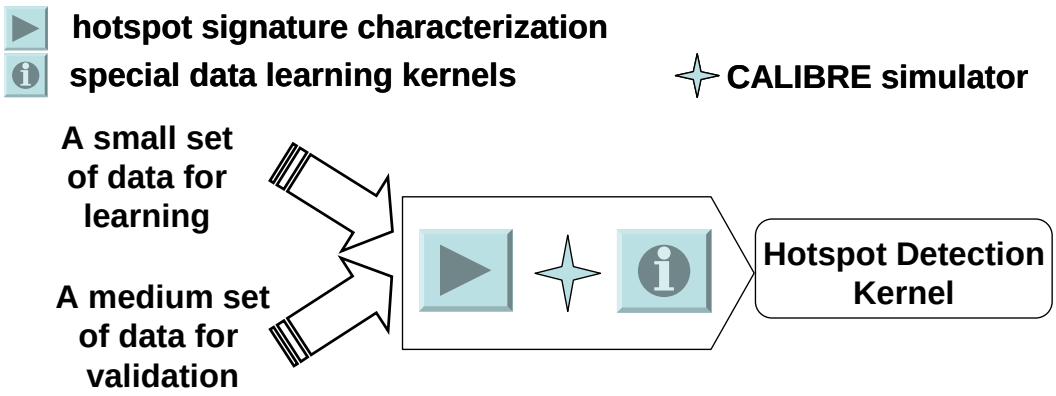


Figure 2.34: Development of the *HD* Kernel

Hotspot Detection technique aims at establishing a compact kernel model to calculate the degree of printability (manufacturability) of certain layout pattern via data mining and classification methods. As shown in Fig. 2.34, we develop our *HD* kernel based on [42], with the following modifications: (1) fine-tuned parameters for enhanced speed and detection accuracy of Support Vector Machine classifier; (2) adjustment of the "hotspot signature metrics" definitions to better take care of line-end and jog characterizations; (3) in-

stead of using an effective radius r, we query the top N nearest neighbor fragments from the database and characterize the context accordingly for better focused meanwhile faster hotspot detection. Please refer to Subsection 2.5.5 for more details regarding the classification and validation accuracies. Please refer to [42] for details of Fig. 2.35 regarding layout fragmentation.

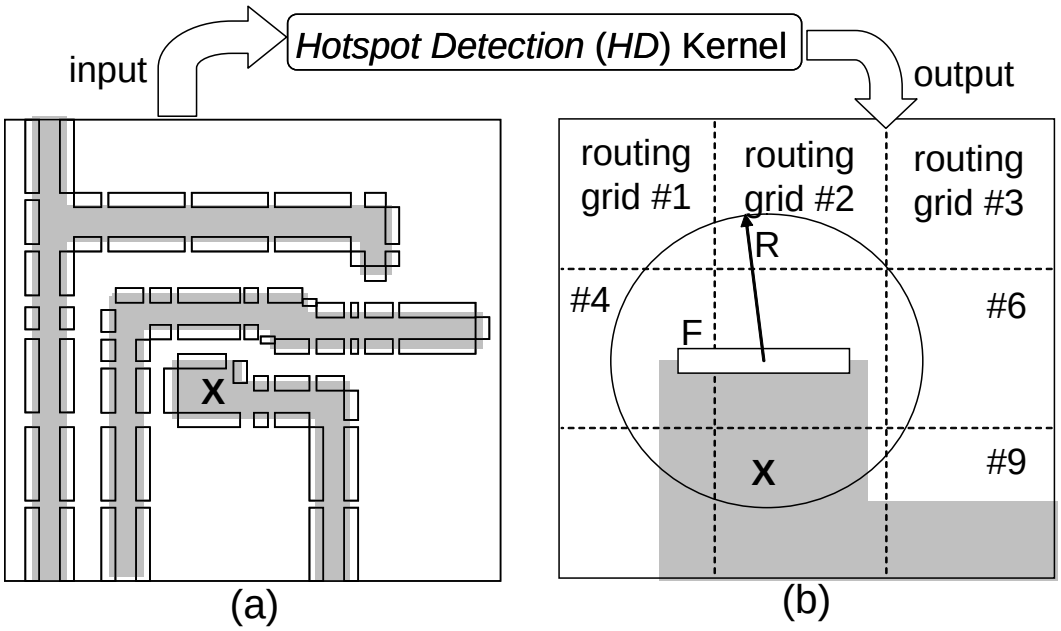


Figure 2.35: Applying the *HD* Kernel for litho-cost update

Once established, the *HD* kernel will be used in the detailed routing stage for calculating $litho(e)^{HD}$ which is the 1st part of $litho(e)$. As depicted in Fig. 2.35(a)(b), *HD* kernel intakes the fragmentation database and returns a quick estimation of layout printability by characterizing the context of fragment F. Since the *HD* kernel is derived *a priori* at a one time cost, it can be used as a quick look-up knowledge base to characterize hotspot conditions

even in the inner design loops.

Since *HD* kernel calculates litho-cost based on database fragments, its result $litho(F)$ must be properly assigned to $litho(e)$ to guide the grid-based detailed routing. Here we use the method shown in Fig. 2.35(b) to map $litho(F)$ onto routing grids. Given an effective radius R, $litho(F)$ is used to update all the unrouted grids that lie within the radius, i.e., grid 1, 2 and 6. Later we will show more details on the fragmentation database update and grid-based cost assignment.

As explained in Subsection 2.5.1, *HD* itself is not sufficient to guide a correct-by-construction router since it only characterizes the contexts of already existing polygons in the layout. We need the *Routing Path Prediction (RPP)* technique to further enhance the lithography-printability on the routes that fall into un-characterized regions.

2.5.4.2 Routing Path Prediction Technique

RPP technique is very important to resolve un-characterized regions and further improve the router's lithography friendliness. We describe in detail the development of *RPP* kernel and its application to calculate $litho(e)^{RPP}$ which is the 2nd part of lithography cost $litho(e)$.

The development of *RPP* involves intensive lithographic computations since it is in nature a greedy searching algorithm. We pre-establish *RPP* at a one time cost and build a multi-objective compact model on top of its knowledge base to allow fast applications inside *AENEID*'s inner design loop.

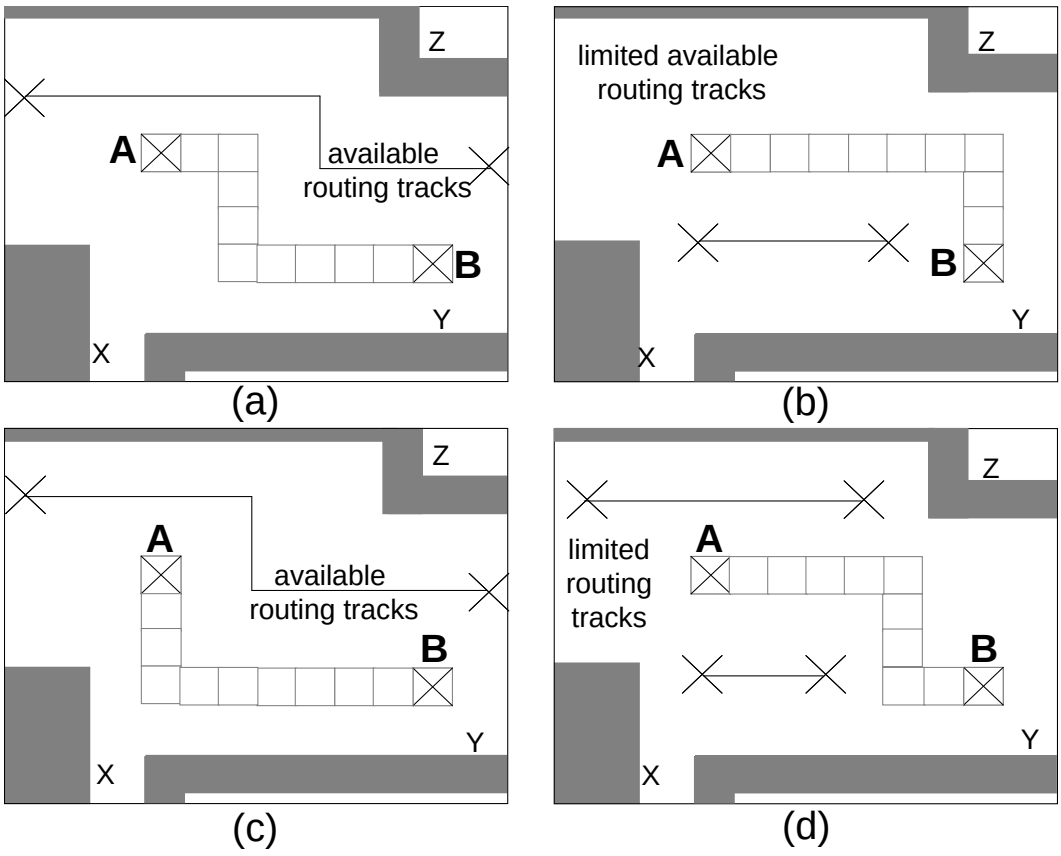


Figure 2.36: A motivational example for lithography-friendly routing path prediction

RPP's optimization objective is given as follows:

$$min\{E[\sum_{j=i+1}^{N} litho(route^j)|route^i]\}, w.r.t \quad i \tag{2.67}$$

where $route^i$ is the current iteration in the detailed router at which point all the routes from 0 to $i-1$ are already fixed. $E[\cdot]$ denotes the mathematical expectation operation. Equation(2.67) aims to find the best route i to take among possible routes i to N, so that the overall potential lithography print-

ability is maximized with possible subsequent routes taken into considerations.

With the example shown in Fig. 2.36(a)-(d), we elaborate the development of *RPP*. Given certain characterized layout context (blockages X,Y,Z) and a pair of pins to route (A,B), we illustrate several possible routes in Fig. 2.36(a)-(d). With each possible route, the number of remaining available routing tracks also varies. In Fig. 2.36(a)(c), there are still available tracks running from left to right side, but for Fig. 2.36(b)(d), all the tracks are limited to local scale. In this case, *RPP* will be established in 3 steps. **Step 1**: explore a wide range of possible routes given the available routing resources. Virtual blockages are placed to help generate a variety of alternative routing paths for Equation.2.67; **Step 2**: run accurate lithographic simulations for all (a)-(d) layout patterns and assess printability; **Step 3**: update two priority queues based on **Step 2** and recommend/encourage a preferable route that: (1) gives the least number of hotspots; (2) provides the most number of available tracks so that subsequent routes can be made easier. Tradeoffs need to be sought if the two queues return different results.

Due to the huge data volume of the resulting knowledge base, we employ a robust neural network classifier to construct the actual prediction model to incorporate into *AENEID*. With the *RPP* kernel ready, we apply it to predict routing paths given the context environment of a net to be routed. In the detailed routing engine, $litho(e)^{RPP}$ lying on the paths that are favored by *RPP* kernel will be adjusted to encourage litho-friendly routing. Thus the $litho(e)^{RPP}$ is updated iteratively inside the router.

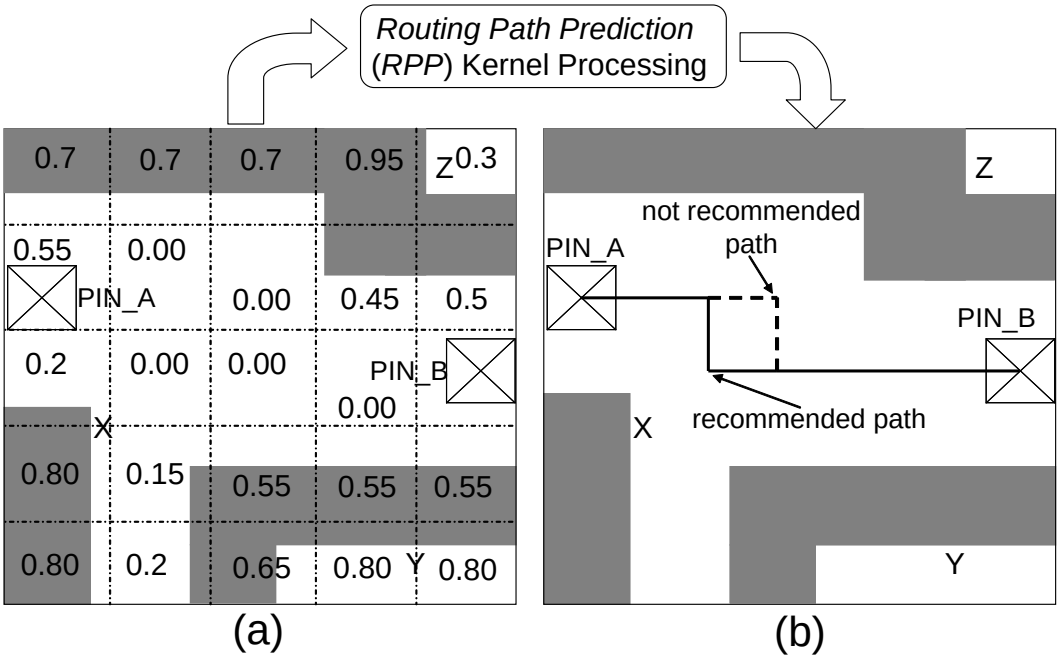


Figure 2.37: The fragment-based litho-cost map update based on *Path Prediction (RPP)*

In Fig. 2.37, we show our layout density based multi-objective neural network model. In Fig. 2.37(a), the input to *RPP* kernel is a vector consists of the pin locations of the net to be routed and the density grid (not routing grid) array whose elements signify blockage densities. *RPP* returns a set of grid locations (region) of preferred routes under lithography-friendly considerations. Based on this information, all the $litho(e)$ touching this path or region are updated accordingly. Please refer to Subsection 2.5.5 for further details regarding kernel training and validations.

2.5.4.3 Fragmentation-based Update

We build our layout database using fragment data structures due to the unique advantages of fragmentation based layout pattern characterization. Here we describe our layout representation techniques in detail.

Conceptually, fragments are defined as vectors lying on the edges of all polygons in the layout. A fragment based layout database makes it very efficient to query entries such as nearest neighbors and polygon width (max distance between internally facing fragments), etc. It also provides satisfactory analyzing resolution (polygon edges) and detection coverage (the whole layout) [42]. Under such a database structure, we use two key techniques for fast data access speed, namely, the *sweep line* algorithm and the *Red-Black Binary Search Tree*.

AENEID requires each fragment to keep updating its neighboring fragments information in real time, since new fragments are introduced into the database as each additional net is routed. Obviously this updating procedure would be invoked frequently whenever the status (occupied/non-occupied) on the routing grid e is changed, thus it needs to be properly taken care of to minimize the runtime degradation. Under such a consideration, we propose a *sweep line* algorithm to obtain the neighboring fragments for a routing path.

Take Fig. 2.38 as an example, assume there are three existing fragments a, b and c on the routing grid shown in Fig. 2.38(a). The arrow appears if two fragments are neighboring to each other. If a new route p is added as shown

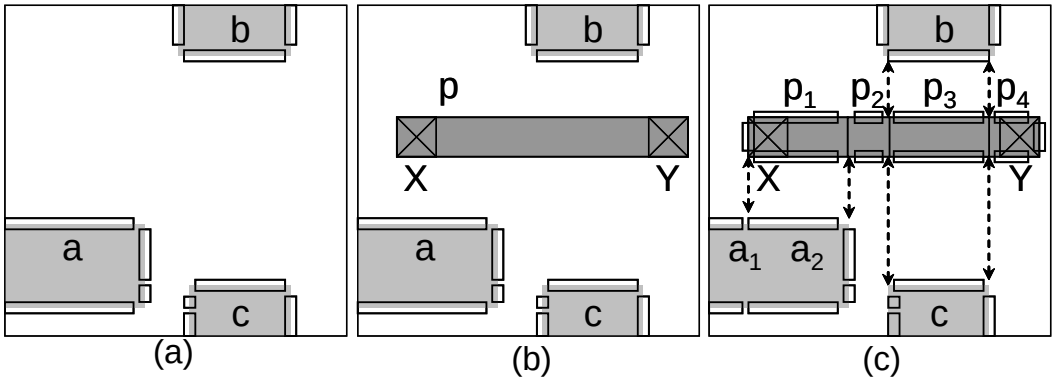


Figure 2.38: Illustration for the fragment database update

in Fig. 2.38(b), all fragments affected by p need to be updated. The algorithm sweeps the routing grid from left to right, each time a new fragment is detected by the sweep line, it checks if there is a pre-swept fragment neighboring to it and do the updating if necessary. Fig. 2.38(c) shows the final result and we can also see that p and fragment a are decomposed to accurately reflect the neighboring situation. By using this algorithm, the updating can be done in $O(\log N)$, where N is the total fragment number in the database. Overall, the effort for the router to interact with our litho-cost related models is $O(k \log N)$ time, where k is the total number of rip-up and route operations.

To effectively update the litho-cost map and identify/query fragments by certain specified layout region, we use an RB-tree to store the information of all fragments. By the property of RB-tree, locating fragments within a given region can be done in $O(\log N)$ time, where N is the number of total fragments in the routing grid. Therefore we can update the litho-cost map in $O(\log N)$ time whenever a path is routed or ripped up.

2.5.5 Simulation and Testing

We implemented *AENEID* in C++ and evaluated it with testing cases in $45nm$ M1-M2 technology process under industry strength Optical Proximity Correction recipes. All simulations are performed on Linux workstations with 4GB memory and Intel Xeon 2.66GHz CPU. In the layout validation baselines, we employed accurate lithographic simulations to locate all the real hotspots based on an edge placement error (EPE) threshold of $8nm$. Inside *AENEID*, we used an aggressive lithography cost upper-bound L to yield the least number of hotspots. *AENEID* can be further sped up by fine-tuning L properly.

2.5.5.1 Training/Validating Machine Learning Models

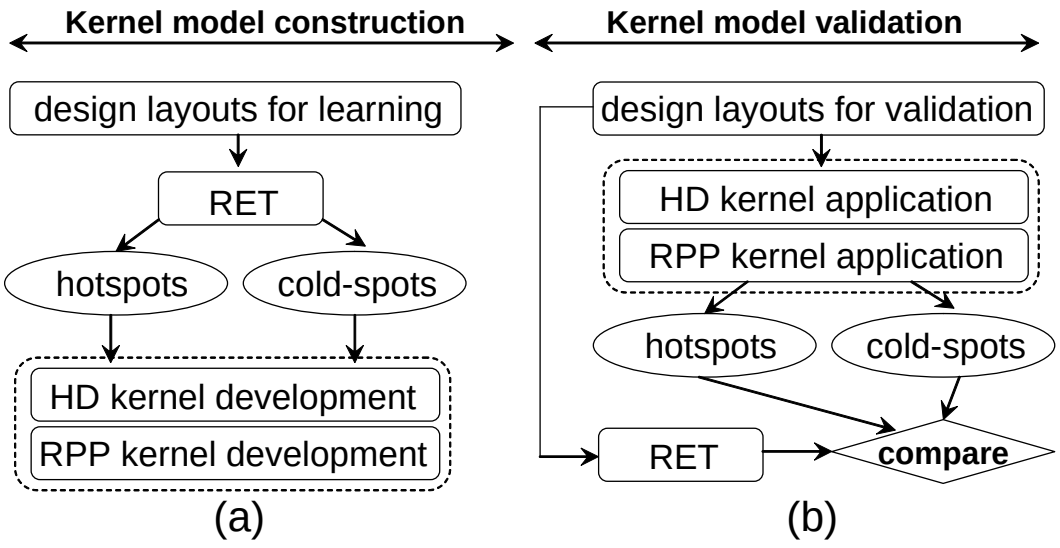


Figure 2.39: Kernel training and validation procedures

Accordingly to the flow shown in Fig. 2.39, we implement the data

learning techniques in C++ and complete the knowledge base update via iterations of training and validation.

For the *HD* model, we use a ν-Class Support Vector Machine classifier. We perform supervised data learning over an 80X80 um^2 design consisting of 40K sample patterns properly labeled by accurate lithographic simulators. Then the established *HD* kernel is validated and tested with various different design samples under the same process technology. *HD* shows 92% of accuracy and 3% of false-alarms for the 40K training patterns, about 88% of accuracy and 8% of false-alarms among another 70K new testing patterns. Such performance is very satisfactory compared with existing studies. The total run-time for the *HD* kernel establishment has a one time cost of about 15 minutes.

For the *RPP* kernel, we first use the proposed greedy search algorithm combined with accurate lithographic simulations. Then we build a multi-objective neural network learning model on top of the derived database using 5 hidden layer neurons and a resilient conjugal gradient learning function with MSE target 0.1. The greedy search is carried out over an 80X80 um^2 design. The neural network model is trained and validated on 200K sample vectors, tested on another 100K samples. *RPP* demonstrates an average 87% accuracy in the training set, and 80% of accuracy in the testing set. Considering the fact that density grids are usually set much larger than the routing grids, *RPP* performs well within its error tolerance. The total run-time for the *RPP* kernel establishment has a one time cost of 3 around hours.

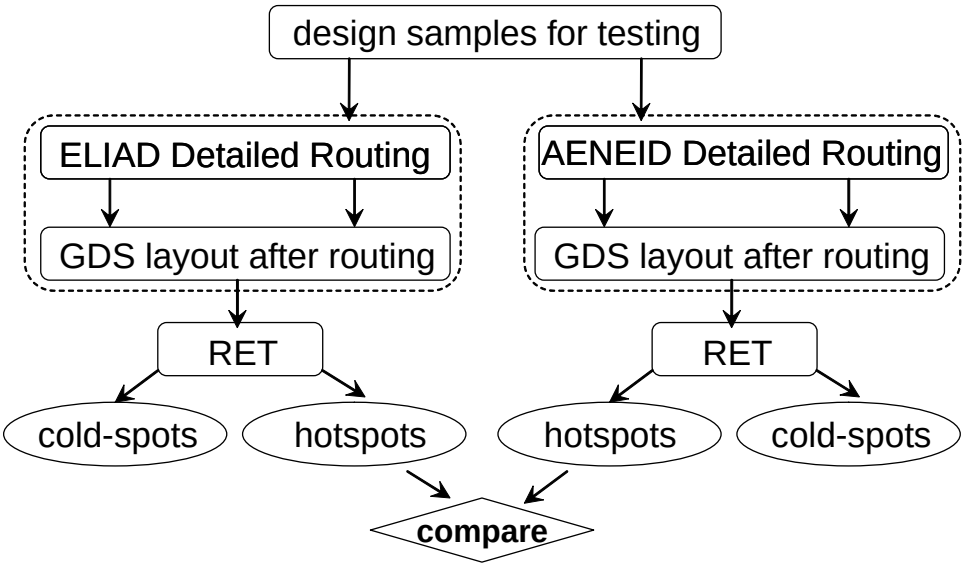


Figure 2.40: The validation flow for ELIAD and *AENEID*

2.5.5.2 Validating/Testing Overall CAD Flow

We test *AENEID* and compare it with [33] according to the flow chart shown in Fig. 2.40, where accurate lithography simulations are employed for hotspots calibration based on $8nm$ of EPE threshold under industry strength OPC setups with $45nm$ M1-M2 process.

Table 2.13 lists 3 industry-strength benchmarks employed to evaluate *AENEID*. These testing cases have not gone through the training or validation process of the *HD/RPP* kernels thus are considered generic and unbiased. Also in Table 2.13 we show the numbers of initial routing blockages and fragments on M1 and M2 layers, which come from the placement of standard cells. Table 2.14 details the experimental result comparisons between ELIAD [33] and *AENEID* in terms of hotspots reduction, total wirelength (WL) and run-time.

Table 2.13: Circuit benchmarks for testing *AENEID*

Benchmarks	CK1	CK2	CK3
Layout Size	50X50um^2	100X100um^2	160X160um^2
Nets to route	0.45K	1.48K	3.4K
M1 Blockage #	1K	8.8K	13.1K
M1 Fragment #	12.2K	41K	152.6K
M2 Blockage #	0.14K	0.47K	2K
M2 Fragment #	0.56K	1.9K	8.3K

In Table 2.14, *AENEID* simulations are run with 2 options in the $litho(e)$ update step: (1) *HD* only; (2) *HD + RPP*. There are several key observations to make in Table 2.14. First, compared with ELIAD, *AENEID* ***HD*** demonstrates about 26%-48% (avg. 36%) hotspot reduction at only 30% of average extra run-time; while *AENEID* ***HD+RPP*** shows 35%-66% (avg. 50%) hotspot reduction at only 29% of average run-time overhead. This shows us: (1) **HD** kernel proves to be compact and accurate than the predictive model used in ELIAD; (2) **RPP** kernel resolves the un-characterized regions thus further reduces the hotpots and improves printability.

Second, **HD+RPP** results in even smaller extra run-time than **HD** alone. This is mainly because the **RPP** kernel in the cost function has reduced the number of rip-up and re-route nets and iterations, consequently ends up saving CPU run-time compared with using **HD** itself.

Third, *AENEID* demonstrates similar total wirelength compared to ELIAD, this is mainly because we employ a similar optimization formulation. This also shows us that **HD** and **RPP** in the cost function did not degrade the total wirelength minimization objective.

Table 2.14: Result comparison between *ELIAD*[33] and our proposed *AENEID*

	ELIAD						**AENEID**											
							HD						**HD + RPP**					
Circuit	CK1		CK2		CK3		CK1		CK2		CK3		CK1		CK2		CK3	
Size um^2	50^2		100^2		160^2		50^2		100^2		160^2		50^2		100^2		160^2	
WL um	860		5509		24789		859		5502		24797		859		5502		24798	
Runtime sec	6		297		2773		8		409		3291		8		400		3279	
Extra time%	-		-		-		33		38		19		33		35		18	
Metal	M1	M2	M1	M2	M1	M2	M1	M2	M1	M2	M1	M2	M1	M2	M1	M2	M1	M2
Hotspot #	17	3	65	10	162	23	11	2	34	7	90	17	8	2	22	5	58	15
# reduc. %	-	-	-	-	-	-	35	33	48	30	44	26	53	33	66	50	64	35
Avg. hotspot reduction %	-						36						50					
Avg. extra run-time %	-						30						29					

(a) ELIAD on CK3 (b) AENEID-HD on CK3 (c) AENEID-HD+RPP on CK3

Figure 2.41: Comparisons of lithography hotspot numbers between ELIAD and AENEID on CK3

In Fig. 2.41, we show the hotspot calibration result visually with: (a) ELIAD on CK3; (b) *AENEID HD* on CK3; and (c) *AENEID HD+RPP* on CK3. Combined with Table 2.14, we conclude that *AENEID* demonstrates greatly enhanced layout printability at acceptable run-time overhead, meanwhile its flow is generic and adaptive to RETs (not only limited to OPC/ORC, etc.). For more break-down details of the simulation, please refer to Table 2.14.

2.5.6 summary

In this section we have proposed *AENEID* - a fast, generic and high performance lithography-friendly detailed router for enhanced manufacturability under advanced process technology, offering manifold advantages: (1) it combines modern data learning methods and novel hotspot prediction techniques to develop compact kernel models through analyzing and learning from a relatively small set of lithography hotspot samples under real industry strength

manufacturing conditions; (2) it applies the pre-established kernels at the detailed routing stage to drive fast high fidelity lithography-friendly interconnect synthesis; (3) its flow and learning procedures are generic to any RETs, not just limited to certain design patterns or OPC setups. *AENEID* is simulated and compared with existing state-of-the-art studies over various industry strength testing cases, demonstrating a significant 22%-66% (50% on average) of hotspot reduction at the cost of only 18%-38% (30% on average) of run-time overhead.

Chapter 3

VLSI CAD for On-Chip Silicon Nanophotonics

Under the current trend of semiconductor manufacturing technology scaling, the traditional IC design and fabrication cycles have become more and more critically challenged by deeper sub-micron (sub-45nm technology nodes) effects, such as lithography variability, leakage currents and the reversed scaling of interconnect signal delay. On the one hand, advanced photolithography has enabled us to make the Complementary Metal-Oxide Semiconductor (CMOS) ICs smaller and cheaper; on the other hand, it also incentives the design and fabrication of new emerging technologies for next generation circuits towards a quantum leap in signal delay and power consumption.

In this chapter, we further leverage the advanced photolithography manufacturing technology discussed in Chapter 2 and study its applications in the monolithic integration of on-chip Silicon-compatible nanophotoincs to assist the design of next generation low-power high-performance opto-electrical IC. The rest of the chapter is organized as follows, in Section 3.1, we introduce a standard cell based methodology to model the basic building blocks of nanophotoincs circuits and systems. We build OIL, an optical interconnect library to assist quantitative physical design explorations for low power in-

terconnect optimization at sub-22nm technology nodes. Section 3.2 presents an optical router for on-chip photonic waveguide synthesis under a low-power driven objective. Section 3.3 presents a hybrid physical design flow for low-power thermal-reliable placement and global routing of Wavelength Division Multiplexing (WDM) interconnect, which is architectured to provide significant improvement of on-chip global routing capacity and signal bandwidth.

3.1 OIL: Optical Interconnect Library

As raised in the International Technology Roadmap for Semiconductors [8], silicon system complexity rockets exponentially due to increasing transistor counts, fueled by smaller feature sizes and insatiable demands for higher integration/performance with low costs. With such aggressive technology scaling, VLSI interconnect effects start playing more and more important roles in the Deep Sub-Micron realm. Below 45nm technology node, traditional copper wire interconnect faces many walls as process scaling leads to various issues such as on/off chip communication bandwidth bottleneck, clock frequency bottleneck, large power dissipation and serious cross-talk noise, etc. To keep up with Moore's Law in the new Tera-bit super computing era, various alternative interconnect techniques [9, 21, 93, 105, 116] have been proposed and analyzed as potential solutions for aforementioned bottlenecks. Among these techniques, optical/photonic interconnect paradigm triggers heated researches (e.g., [9, 13, 29, 54, 74, 123]) and is considered as a potential quantum leap towards next generation VLSI on-chip interconnect technology.

The idea of introducing optical interconnect onto integrated circuit chips was first proposed by [51] in 1984. Although optical fiber enabled long-haul photonic interconnect started webbing the globe since the 90's, it was only until more recently that intra/inter IC chip level photonic interconnect researches truly took off. On PCB level, [128] proposed a fully embedded board-level optical interconnect schematic from OWG (optical waveguide) fabrication to device integration. For inter/intra chip communications, various high performance photonic devices have been researched and developed in both academia (e.g., [86, 94]) and industry (e.g., [54, 68, 123]). EDA based physical synthesis flows for on-chip optical interconnect planning [44, 87] have also been published; together with new architectures for high performance on-chip photonic interconnection. One important architecture is the photonic Networks-on-Chips paradigm [57, 109], where data packets are routed on a on-chip photonic network with high speed photonic interconnects shared in a Time Division Multiplex (Wavelength Division Multiplex) manner.

As projected by [8, 22, 86], optical interconnect outperforms traditional Cu/Low-K interconnect with significant potentials as technology scales down, in terms of high through-put, small propagation delay, low power consumption and low soft error rate, etc. On-chip optical interconnect also demonstrates promising potentials compared with carbon nanotube bundle interconnection [29] in terms of power dissipation and communication latency/bandwidth. Geared up by the recent advances in silicon nanophotonic (e.g., [55, 123, 125]), it is a good time for design space explorations for CAD and architecture level

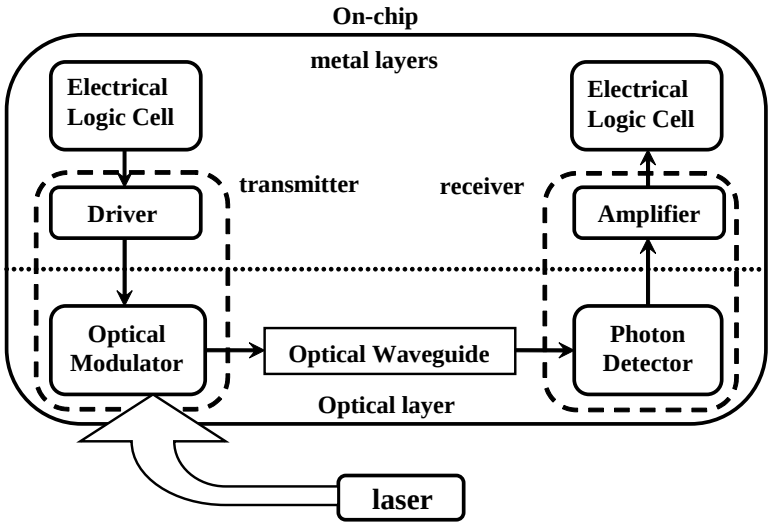


Figure 3.1: Block diagram for Optical-Electrical and Electrical-Optical data conversions using an off-chip laser source

optimizations towards large scale on-chip nanophotonics integration.

3.1.1 Related Work and Our Contributions

On-chip nano-photonic interconnect consists of silicon optical waveguide and opto-electrical/electro-optical conversion devices as shown briefly in Fig. 3.1. For the past few years, researches for on-chip nano-photonic integration mainly focus on two aspects: device level fabrication (e.g., [54, 55]) and network architecture implementation (e.g., [108, 114, 127]). On device fabrication level, various nano-photonic Giga-scale modulators (e.g., [52, 55, 85, 125]), photo-detectors (e.g., [23, 95, 104]), couplers, switches (e.g., [96, 124]), buffers, on-chip waveguide and on-chip WDM (Wavelength Division Multiplex) devices (e.g., [13, 58, 59]) have been demonstrated in both industry and academia. On

architecture level, intrigued by Network-on-Chip paradigm, many new on-chip photonic architectures are proposed (e.g., [61, 108, 127]), together with novel network packet routing mechanisms [74, 108] and performance analysis [19]. CAD based performance driven synthesis for on-chip photonic integration has also been proposed, such as timing-driven on-chip optical waveguide routing for 3D system-on-package [87].

To further leverage the photonic Network-on-Chip paradigm for future generation Chip Multi-Processors, we first establish OIL: a parameterized library for low-power on-chip photonics integration CAD exploration, utilizing a collection of silicon compatible nano-photonic devices built on silicon-on-insulator. OIL (Optical Interconnect Library) allows us to quantitatively explore CAD optimization methods for on-chip photonics synthesis on system level in terms of power consumption and communication latency, etc., under various data constraints imposed by the device characterizations. To apply OIL, we present a *new Photonic Networks-on-Chip* architecture, incorporating within-core optical interconnect planning and core-to-core optical network routing onto a single layer for enhanced photonic silicon utilization.

The rest of this section is organized as follows, Subsection 3.1.2 gives a detailed description for Optical Interconnect Library, followed by Subsection 3.1.3, a new architecture for photonic on-chip communications. In Subsection 3.1.4, OIL is applied to evaluate our proposed architecture, in terms of performance improvement, power consumption, insertion loss and performance scalability. Subsection 3.1.5 concludes the section with a brief summary.

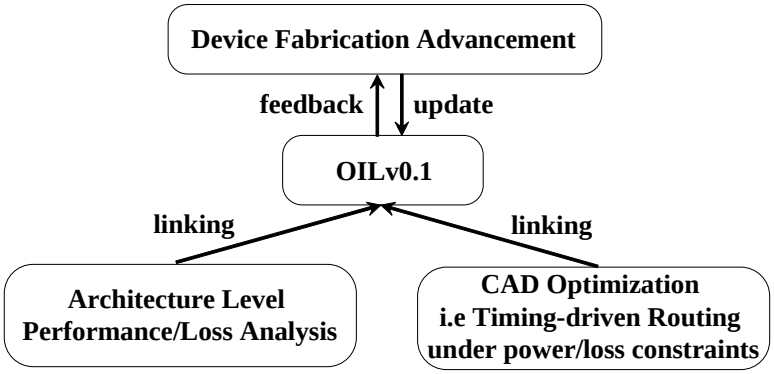


Figure 3.2: OIL - A nanophotonic Optical Interconnect Library for on-chip photonic integration analysis/optimizations

3.1.2 Optical Interconnect Library

After a comprehensive study of the current pool of photonic devices, we establish a standard cell library OIL: an Optical Interconnect Library of on-chip nanophotonic components. OIL is an extensible set of devices including optical modulators, photodetectors, buffers, switches, couplers, optical waveguide and on-chip WDM devices. Based on recent advances in the nano-photonics fabrication, OIL can contribute to a close connection between the device fabrication, architecture design and CAD optimization towards a promising on-chip photonic integration solution, as illustrated in Fig. 3.2. For more details regarding OIL and future updates, please refer to [7].

3.1.2.1 Nanophotonic Modulators

Nanophotonic modulators are used for on-chip electrical to optical data conversion. Under photonic networks-on-chip architecture, a modulator is to be inserted at each gateway (G) on every processing unit of a chip multi-

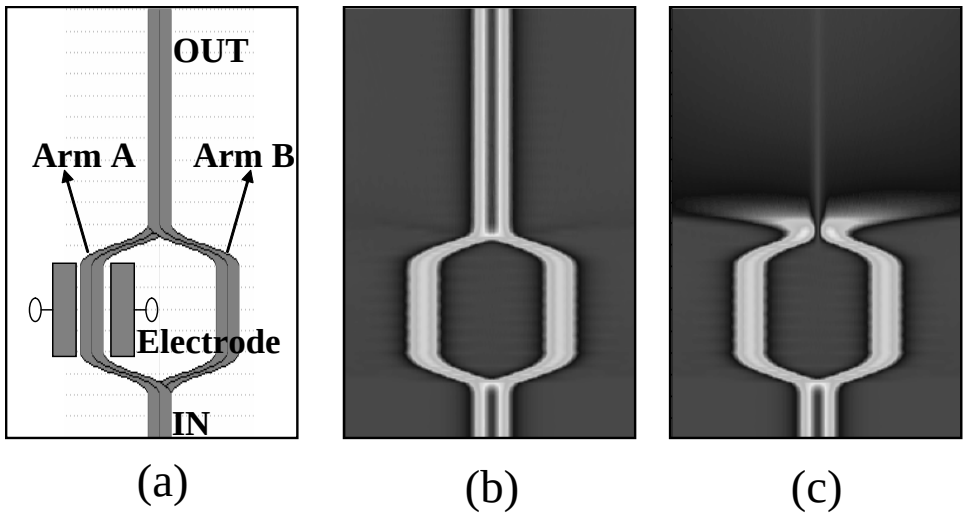
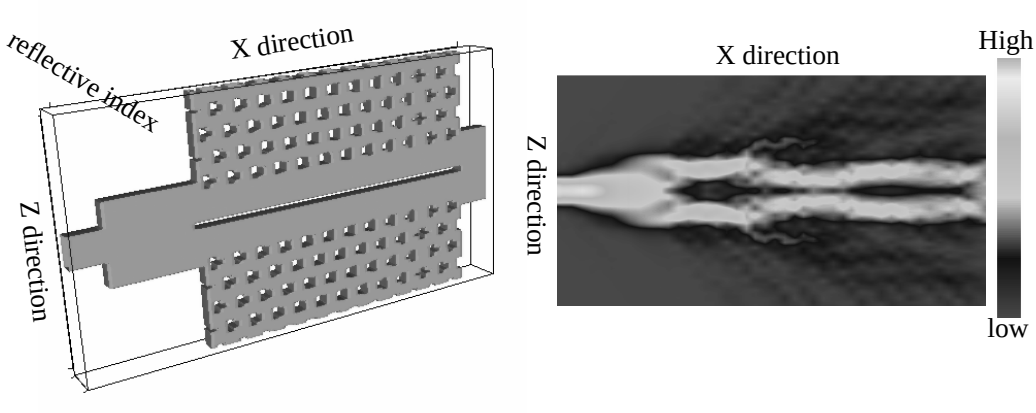


Figure 3.3: (a) Working mechanism for a Mach-Zehnder photonic modulator, with modulation ON state in (b) and OFF state in (c), where state switching is controlled by electrode voltage

processor. Current nanophotonic modulators in OIL fall into two classes: Mach-Zehnder structure modulator (e.g., [53, 55, 81, 107]) and ring resonator structure (e.g., [85, 136]).

Mach-Zehnder Modulator

The working principle of a typical Mach-Zehnder modulator is briefly illustrated in Fig. 3.3, where the reflective index of arm A is manipulated by control voltages on the electrode, leading to a phase modulation of the optical wave propagating through arm A. ON condition is depicted in Fig. 3.3(b) when the phase shift in arm A is integer times of 2π. In Fig. 3.3(c), the control voltage results in $n \cdot \pi$ (n is odd integer) phase shift, causing an OFF state on the output port. A modified arm with photonic crystal structure in



(a) 3D device index profile (b) BeamPROP simulated optical wave |E|

Figure 3.4: A modified arm design for Mach-Zehnder nano-photonic modulator in OIL based on [55], where (a) is the arm design using photonic crystal OWG on Silicon-on-Insulator; (b) is the Rsoft [6] simulated optical wave electrical field amplitude spectrum

OIL is visualized in Fig. 3.4(a) with its simulated electrical field amplitude spectrum in Fig. 3.4(b).

Ring Resonator Modulator

Ring resonators modulate optical signals by selectively coupling signals from the optical waveguide, the selected wavelengths are defined as the resonant wavelengths of the ring. Similar to Mach-Zehnder modulator, such resonant wavelength is generally controlled by applying voltages across the ring structure.

A typical micro ring structure resonator is shown in Fig. 3.5 with OFF state FDTD simulation in Fig. 3.5(b) and ON state in Fig. 3.5(c), transient behaviors of corresponding add/drop port are depicted in Fig. 3.5(d)(e).

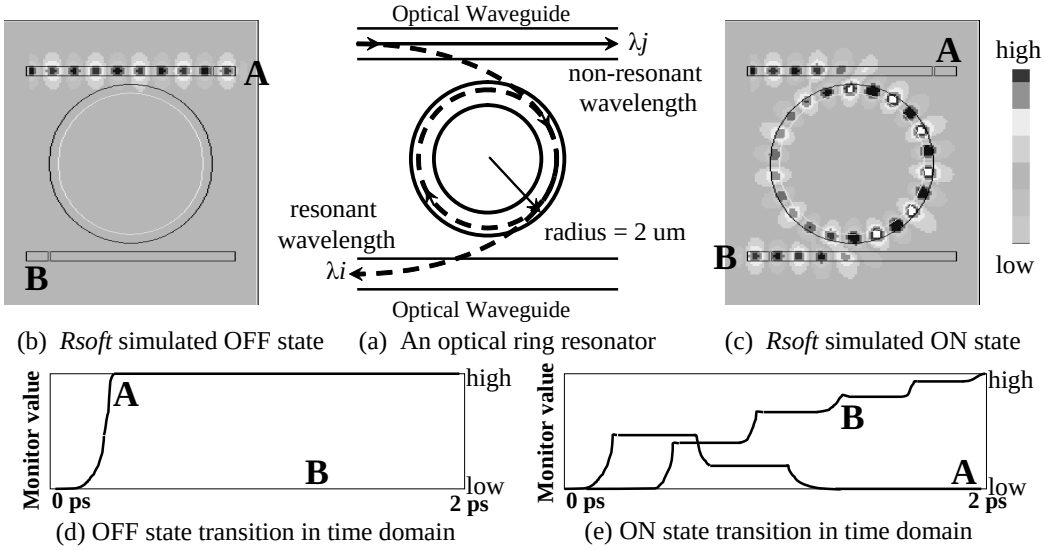


Figure 3.5: (a) A ring resonator in OIL based on [93]; (b)(c) are FDTD simulated results in ON and OFF state respectively using [6]; (d)(e) are corresponding transient waveforms on port A and port B

Table 3.1 summarizes some high level parameters of optical modulators in OIL, from which we can learn the basic trade-offs for different devices in terms of bandwidth, throughput, loss figure and footprint area, etc. Currently demonstrated Mach-Zehnder structure modulators outperform ring resonators in terms of throughput and bandwidth limit, yet taking up much larger footprint areas, and suffering from more on-chip insertion loss. Optimal selection of on-chip optical modulators becomes essentially CAD optimization problem under various on-chip resource constraints (area, density, loss and/or performance specifications).

Table 3.1: High level parameters of on-chip nano-photonic modulators

Mach-Zehnder Optical Modulator					Ring Resonator Modulator		
	Mod1_MzM	Mod2_MzM	Mod3_MzM	Mod4_MzM		Mod5_Ring	Mod6_Ring
length	100 um	>1000 um	<5 um[a]	80 um	length	40 um	15 um
width	<10 um[d]	<40 um[d]	6 um	<20 um[d]	width	30 um	12 um
speed[a]	10 Gb/s	30 Gb/s	1.0 Gb/s	>1.0 Gb/s[g]	modulation rate	>1.0 Gb/s[g]	12.5 Gb/s
power[b]	5.1 pJ/bit	600 mW[e]	<200mW[f]	–	bending radius	5 um	5 um
loss[c]	12 dB	7dB	21.3dB	8.18dB[g]	ring loss	0.07 dB[g]	<0.45 dB[d]
λ_0	1550 nm	1550 nm	∼1568 nm	1541 nm	resonant wavelength	1550 nm	1558.86nm

[a] Speed reported is the modulation rate in the unit of Gb/s.

[b] Power consumption reported accounts for the total power consumed, including optical power for the signalling and electrical power for biasing the circuits.

[c] Loss reported is the total on-chip optical power loss in dB unit.

[d] Estimated or simulated based o on: [53], [81], [107], [55], [85], [136], [70], [139], [23], [95], [134], [14], [59], [84], [124].

[e] Bias power, estimated from [81].

[f] RF power consumption for 10GHz signal.

[g] Theoretical projections or device simulation results using [6].

Table 3.2: High level parameters of on-chip nano-photonic photo-detectors

On-chip photo-detector					
	Detector1_GOI[a]	Detector2_Gnip[a]	Detector3_Gnip[a]	Detector4_GOI[a]	Detector5_Ge[a]
footprint	10X10 um^2	7.4X50 um^2	4.4X100 um^2	10X10 um^2	<10X10 um^2
BW	29GHz	31.3GHz	29.4GHz	40GHz[d]	39GHz
Bit rate	50 Gbps	40 Gbps	40 Gbps	>40 Gbps[d]	40 Gbps[d]
wavelength	850,895nm	1550nm	1550nm	1527nm	1552nm
quantum efficiency	40%	71%	93%	>90%	-
operating voltage	1.0V	5.0V	2.0V	<4.0V	2V
dark current	<24nA	169nA	267nA	100nA	75nA for 1V bias

[a] Estimated or simulated based on: [53], [81], [107], [55], [85], [136], [70], [139], [23], [95], [134], [14], [59], [84], [124].
[b] Bias power, estimated from [81].
[c] RF power consumption for 10GHz signal.
[d] Theoretical projections or device simulation results using [6].

3.1.2.2 On-chip Photodetectors

Photodetectors perform the function of optical to electrical data conversions at each terminal node of an optical path. Being the last component on a photonic path, there are several key parameters to characterize for a detector: *detecting bit rate* serves as an important constraint for photonic communication link design because it imposes an upper-limit to the otpical layer data throughput; *power consumption* is also crucial since detectors are used in large quantity for high fan-out nets; *photo-detection power threshold* is another key constraint for low power driven CAD optimizations. Under such a constraint, optical modulators and waveguide must be planned optimally to guarantee successful optical-electrical conversion at each terminal node (sink).

From Table 3.2 we learn that current optical detectors provide fairly high throughput for optical to electrical data conversion with relatively small footprint area. However, ultra low power detector with smaller footprint is desired for high density on-chip photonic integration since photodetectors are present on optical links in large quantity.

3.1.2.3 Switches, Couplers and Buffers

On-chip Nanophotonic Switch

On-chip nanophotonic switches can be employed to achieve Division Multiplex functions and can be used for constructing core-to-core photonic networks for a chip multi-processor. There are various ways to implement an optical switch, Fig. 3.6 shows a design of 1/8 switch constructed by seven 1/2

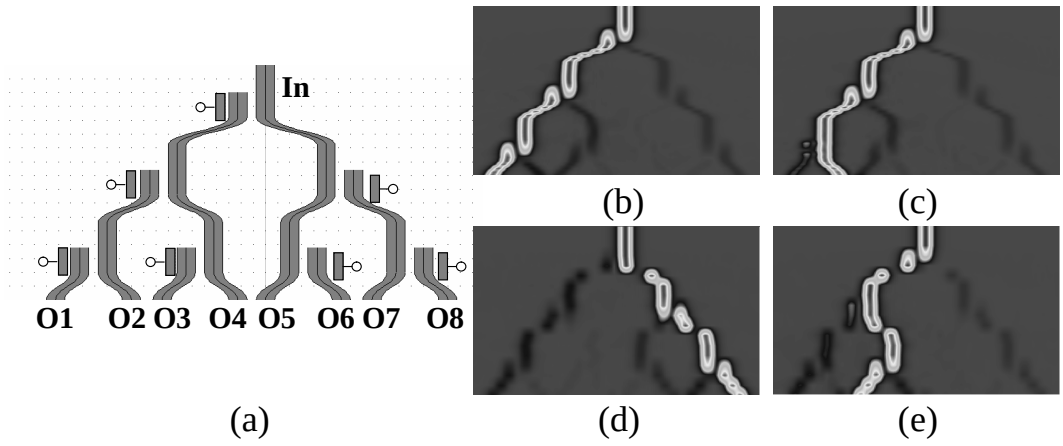


Figure 3.6: (a) A 1/8 transport switch array built with *Switch3_Trans* in OIL; (b)-(e) are simulated results under different electrode voltages using [6]

transport switches from OIL. Ring resonators from Fig. 3.5 are also favored for utilizing switching / re-directing functionalities due to compact footprint and relatively low insertion loss.

On-chip Nanophotonic Coupler

Nanophotonic switches and couplers are important devices for our proposed *holistic photonic NoC* architecture since they make the within-core waveguide routing possible in a *Gridless Single Layer with Coupling* manner. Shown in Fig. 3.7 is the working principle of an optical coupler. Under Optical Amplitude Modulation, an optical coupler can be configured to deliver the following functions: PortA $\equiv$ PortD, PortB $\equiv$ PortC with a certain amount of insertion loss, enabling great flexibility for optical routing exploration. In Table 3.3, we give several types of switches and couplers, which we will use to evaluate network router loss figure in Subsection 3.1.4.

Table 3.3: High level parameters of on-chip nano-photonic switches, rings, couplers

Micro-ring based switch			Transport switch		Optical coupler	
	Switch1_Ring[d]	Switch2_rings[a]		Switch3_Trans[d]		Coupler1[d]
ring radius	1.8um	4um × 5rings	length	∼10 um	length	∼15 um
coupling gap	0.2um	∼0.2 um	width	2.5um	width	<5 um
passing loss	<0.01[d] dB	<0.3[a] dB	coupling loss	<0.02 dB/coupling[d]	coupling loss	<0.35[d] dB
coupling loss	<0.5 dB[d]	<2.5 dB[a]	OWG bend loss(r=3um)	<0.02 dB/bending[d]	OWG bend loss	<0.05 dB/bend[d]

[a] Based on: [53], [81], [107], [55], [85], [136], [70], [139], [23], [95], [134], [14], [59], [84], [124].
[b] Bias power, estimated from [81].
[c] RF power consumption for 10GHz signal.
[d] Theoretical projections or device simulation results using [6].

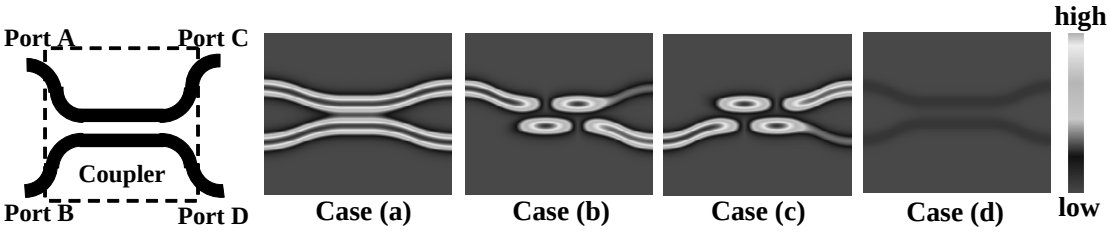


Figure 3.7: Working mechanism for an optical coupler simulated with [6]

On-chip Nanophotonic Buffer

Nanophotonic buffers (Table 3.4) contribute to on-chip photonic signal delay/buffering for some special purposes. Up to now, nanophotonic buffers are not commonly used for on-chip applications, since packet switching based core-to-core communication operates in a globally asynchronous manner and there is no sequential logic functions on the photonic Network-on-Chip layer that require buffering for timing requirements, etc.

An FDTD simulated buffer with 5 stages of coupling ring switches from OIL is shown in Fig. 3.8, with a plot of its transient behavior on the bottom.

Table 3.4: High level parameters of nano-photonic buffers

Optical buffer		
	Buffer1_APF[a]	Buffer2_CROW[a]
footprint	0.09 mm^2	0.045 mm^2
ring radius	6.5 um	6.5 um
ring number	56 APF	100 CROW
coupling gap	0.2 um	0.2 um
insertion loss	22dB	23dB
buffer cap	10bits at 20Gbps	1bit at 5Gbps

[a] Estimated or simulated based on: [134] and [6].

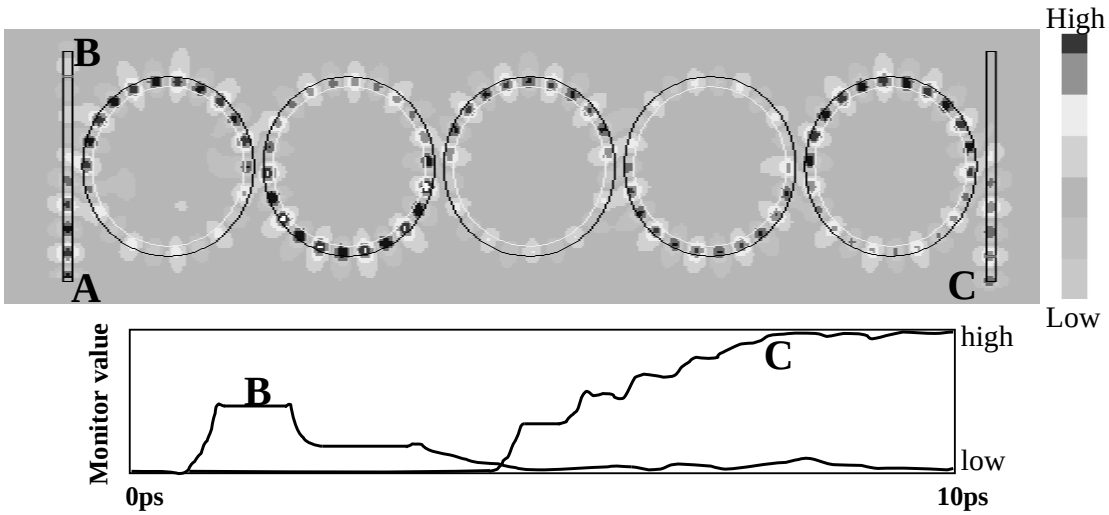


Figure 3.8: Above: A ring-switch based nanophotonic buffer in OIL; Below: FDTD simulation for the on-chip optical buffer using [6].

3.1.2.4 On-Chip Optical Waveguide

Under *Grid-Less Single-Layer Routing with Couplings* (details in [44]), within-core optical waveguide routing becomes very flexible. To characterize an optical path, OIL defines 3 types of waveguide losses in equations 3.1- 3.4, as illustrated in Fig. 3.9, where P_{loss} is waveguide propagation loss, it is proportional to the length of optical interconnect, with a coefficient α; B_{loss} is the bending loss, it is related to the degree of the optical interconnect (silicon waveguide) bending arc angle θ, and the radius r of the bend; C_{loss} is the coupling loss, proportional to the number of couplers (crossings) on the interconnect, with a coefficient γ in dB.

$$Total_{loss} = P_{loss} + B_{loss} + C_{loss} \tag{3.1}$$

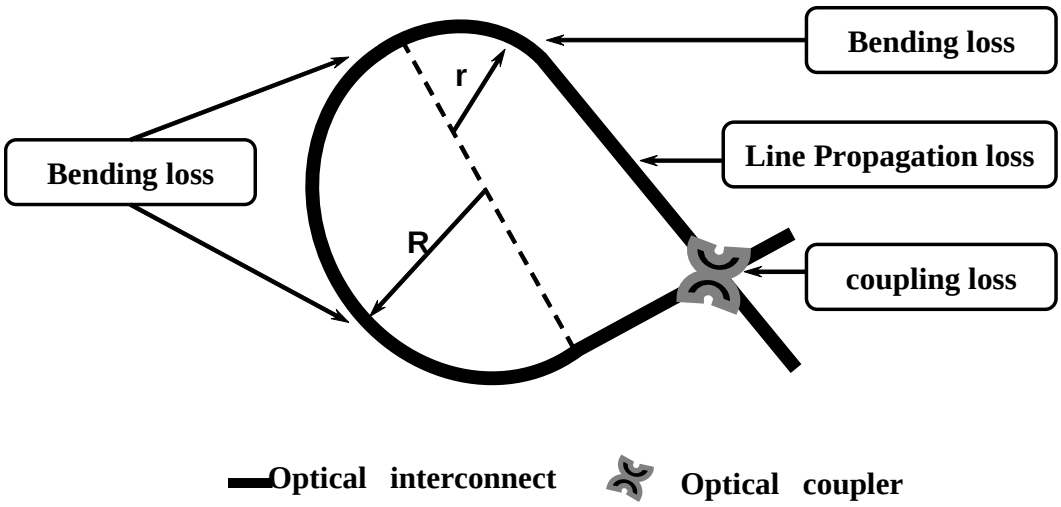


Figure 3.9: Sources of loss for on-chip photonic waveguide

$$P_{loss} = \alpha \cdot length_{path} \tag{3.2}$$

$$B_{loss} = \beta \cdot f(\theta) \cdot g(r) \tag{3.3}$$

$$C_{loss} = \gamma \cdot Num_{couplers} \tag{3.4}$$

Fig. 3.10 plots the simulation results of total insertion loss on a bending waveguide using [6]. With small bending radius (<30um), bending waveguide sidewall surfaces serve as dominant sources of total insertion loss $Total_{loss}$; as the radius gets larger, bending loss B_{loss} decreases to zero and waveguide propagation loss P_{loss} becomes the major source of loss. In OIL, α is set to $1.5dB/cm$ for optical waveguide with 450 nm width and 230 nm thickness silicon core on insulator, to $4.5dB/cm$ for optical waveguide with 200 nm width and 100 nm thickness silicon core ($n \simeq 3.46$) on silicon dioxide ($n \simeq 1.46$). OIL calculates $Total_{loss}$ of small radius bending waveguide with table look-up

and interpolation methods based on device simulation using [6]. For $Total_{loss}$ calculation of large bending radius waveguide, equation 3.2 is adopted as a close enough approximation [93].

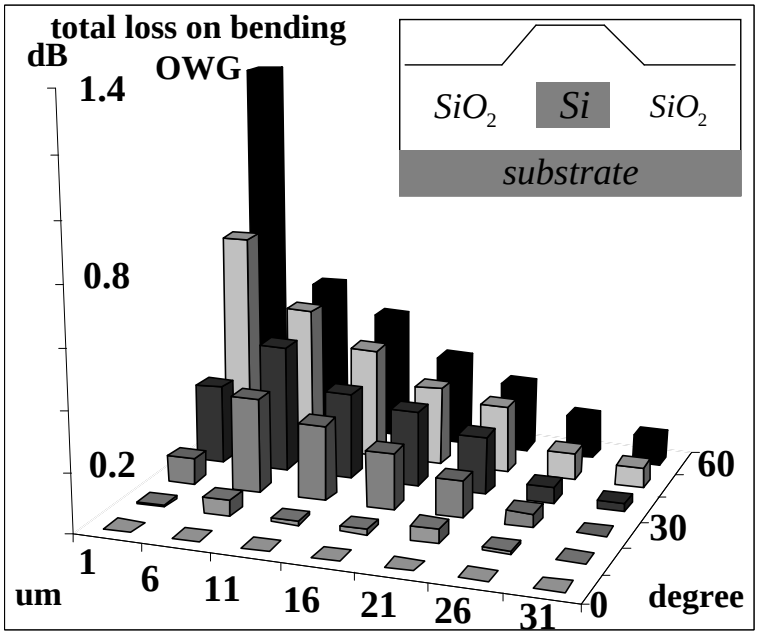


Figure 3.10: Total simulated insertion loss on a certain bending optical waveguide (200nm wide, 100nm thick, $n \approx 3.5$) with small bending radius (1um−31um) and small bending degree (0−60degree) in OIL using [6]

3.1.2.5 WDM On-Chip

WDM (Wavelength Division Multiplex) technique has been playing an active role in long haul optical communication for a long time. In WDM, multiple signals are modulated by different wavelength light beams and transmitted through a single multi-mode fibre (or via free space) in a wavelength multiplex manner. Although there are still major challenges to be properly

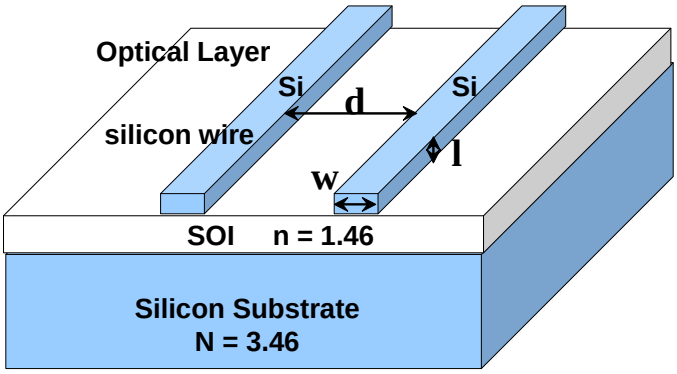


Figure 3.11: On-chip optical waveguide on Silicon-on-Insulator

addressed before it becomes a viable application for on-chip scale, latest device fabrication advancements such as [13, 14, 59, 75] demonstrate promising potentials for major break-throughs in the near future.

Table 3.5 shows a few WDM related devices in OIL with some key high level parameters. While on-chip WDM device design and fabrication faces major challenges, it holds essential potentials for high throughput photonic Networks-on-Chips.

Table 3.5: High level parameters of on-chip WDM devices

On-chip WDM (Mux/Demux, Add/Drop filter)			
	WDM1_AWG[a]	WDM2_AWG[a]	Filter_MZL[a]
footprint	<0.6 mm^2	0.1mm^2	<0.02mm^2
spec	1to4/4to1	1to16/16to1	add/drop
cross-talk	<-20dB/10Gbs	-20dB	-10dB
insertion	<2dB	2.2dB	<2.5dB

[a] Same as in Table 3.2

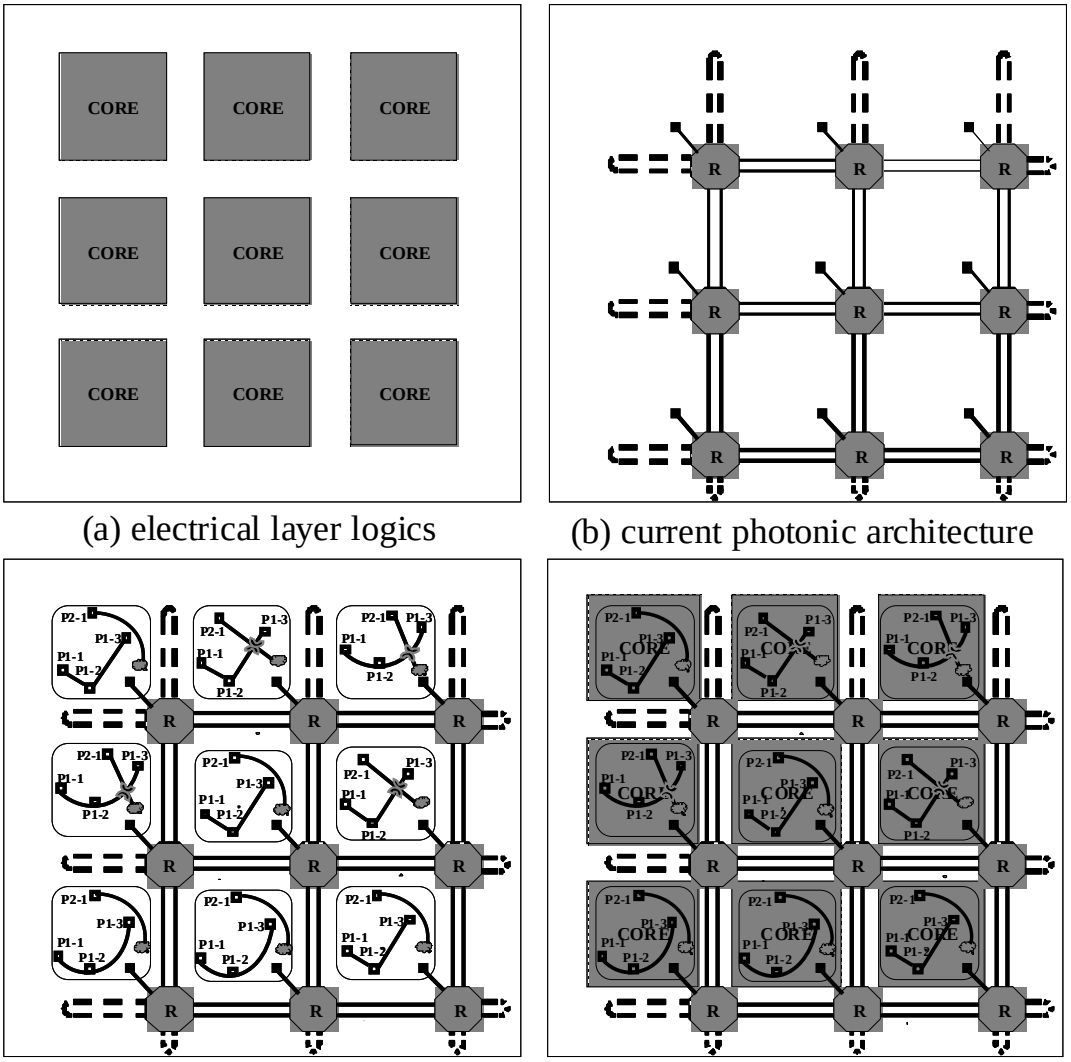
3.1.3 A Holistic Photonic Network-on-Chip

Network-on-Chip related architectures arose as a special class of applications for chip multi-processor communication efficiency, where high speed electrical wires are shared in Time Division Multiplex manner on a dedicated electrical network for core-to-core data packet routing, etc. Despite of its many advantages there is no true relief of on-chip power dissipation [12] for such an architecture on the electrical layer. Photonic NoC preserves the advantages of electrical NoC, meanwhile demonstrating great resilience in terms of higher bandwidth/throughput and low power consumption on a silicon photonic layer.

Based on existing photonic NoC, our proposed architecture combines *photonic waveguide routing* and *network routing* together onto a dedicated on-chip optical layer for improved performance and enhanced silicon utilization. Such a regulated architecture allocates the photonic layer resources in a systematic manner thus contributing towards sustainable high density on-chip photonic integration for future technology nodes. In the long run, the employment of a dedicated photonic silicon layer contributes towards high *area utilization*, *mask reusability* for existing CMOS silicon/metal layers and high *flexibility* for photonic layer design explorations.

3.1.3.1 Architecture Overview

Fig. 3.12 gives an architecture overview of a chip multi-processor that is divided into hierarchical layers. They are employed for digital logics (heterogeneous IP cores), global communications and on-chip memories, etc.



(a) electrical layer logics

(b) current photonic architecture

(c) proposed photonic architecture

(d) Overview for electro-optic layers

Figure 3.12: (a) traditional multi-core processor on electrical layers; (b) a general case of previously proposed Photonic Network-on-Chip architecture on SOI; (c) our proposed *New Photonic Networks-on-Chip* architecture combining (b) and within-core optical routing scenarios onto a dedicated photonic silicon layer; (d) a top-level view of the combination of (a) and (c)

As illustrated in Fig. 3.12, Fig. 3.12(a) is the electrical layer with heterogeneous IP cores; Fig. 3.12(b) shows a typical photonic Networks-on-Chips (NoC) architecture. Although a non-blocking photonic NoC requires some extra infrastructures, the overall occupancy of photonic silicon layer is still low under such an architecture since only a small portion of it is utilized. Based on such an architecture, our approach aims at further improving the photonic silicon utilization towards whole chip performance enhancement.

An overview of our proposed holistic architecture on the photonic layer is shown in Fig. 3.12(c), and the top-view of the whole chip is shown in Fig. 3.12(d) with the electrical layers stacked at the bottom of our proposed photonic network. Such a new architecture is composed of two main parts:

- A global photonic routing network for efficient core-to-core communication with links shared in Time Division Multiplexing and Wavelength Division Multiplexing.
- A set of within-core optical waveguide routings for a properly selected set of nets on the low latency photonic layer, for timing performance improvement of each precessing unit.

The first part aims at the design of high throughput / bandwidth core-to-core communication with low power consumption, while the second part must be properly supported by an optimized CAD flow for performance and/or power driven objectives, subject to various constraints parameterized by OIL.

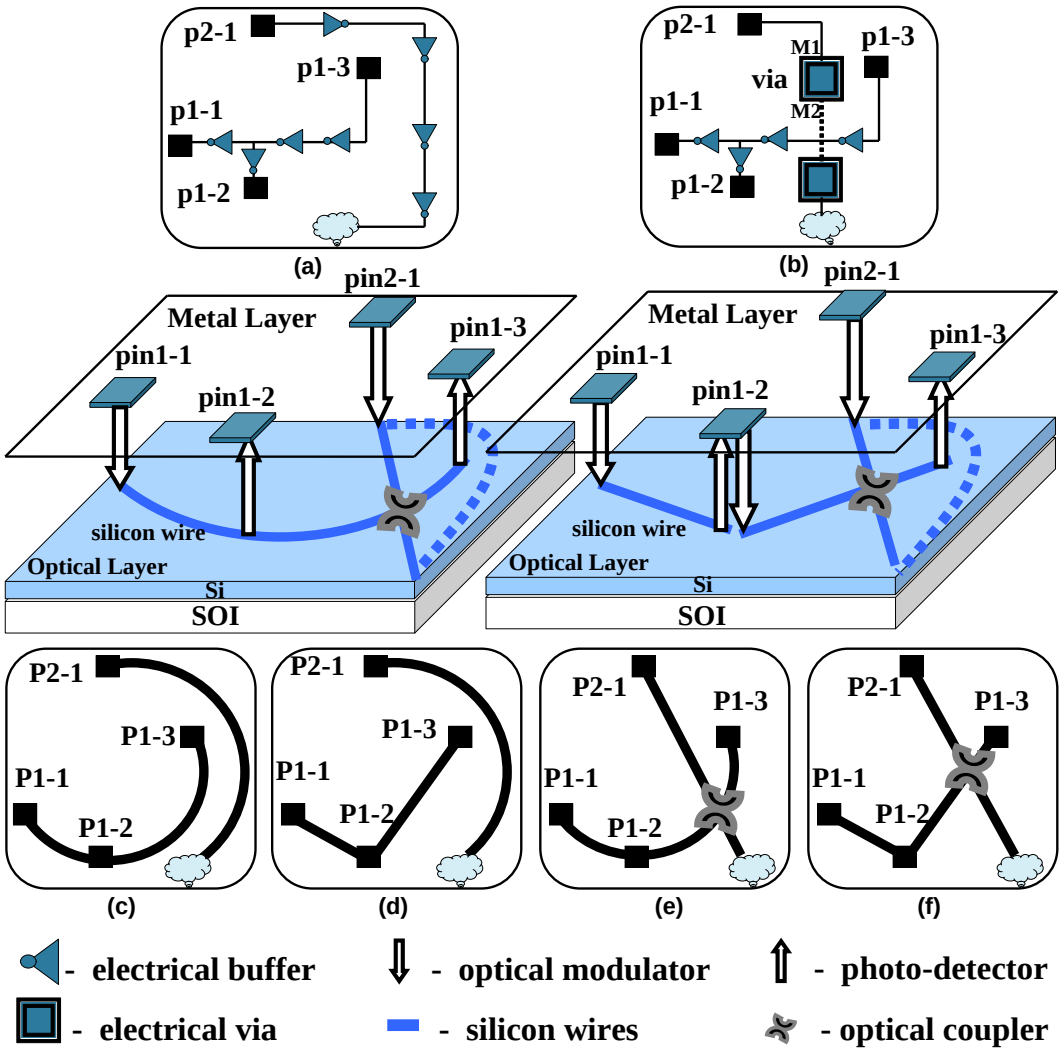


Figure 3.13: Illustration for within-core photnic interconnect planning v.s traditional electrical wire planning, where (a)(b) are two possible electrical routing scenarios for pin1-1/2/3 and pin2-1; (c)(d)(e)(f) are four possible optical routing choices on the potonic layer for the same pins

3.1.3.2 Wire and Packet Routing

On-chip optical interconnect offers unique characteristics when compared with traditional copper-based interconnect in many aspects, such as improved power dissipation and low signal propagation latency, etc. While RC delay for a copper wire increases quadratically with wire length, photonic interconnect latency maintains a linear increase with a constant group velocity. In the following subsections, we describe our new photonic architecture that combines *optical wire routing* (with-in processing unit) and *optical packet routing* (between processing units) as a holistic approach for chip multi-processor performance improvement for next generation super computing.

Within-Core Optical Wire Routing

With the unique properties of on-chip photonics, we propose a *Gridless Single Layer with Coupling* based optical routing technique for the optical netlist. Our optical routing rationale is illustrated in Figure. 3.13, where there are 2 nets to be routed within a core, noted as *pini-j*, meaning it is the *jth* member of net *i*. Fig. 3.13(a) and (b) shows two alternatives for conventional routing on electrical layer with buffers and/or metal via inserted to alleviate the timing penalty caused by the long wires across the chip. Buffers are inserted since RC delay increases quadratically with electrical wire length. Yet buffer insertion is not all-powerful and global timing nets are generally hard to close with traditional copper wires.

Fig. 3.13(c)-(f) show 4 possible routing geometries for the 2 nets on

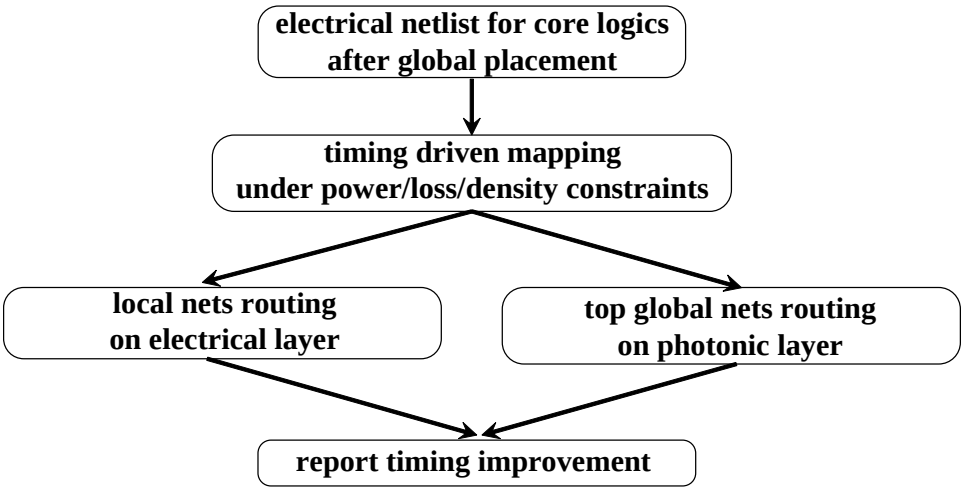


Figure 3.14: A CAD flow for optical-electrical co-synthesis to improve circuit performance using within-core optical interconnect.

optical layer based on our optical routing, where nanophotonic devices such as modulators, photodetectors, couplers and optical waveguide are integrated on a silicon photonic layer, in the presence of coupling loss, waveguide bending loss and photodetection threshold constraints etc. Coupling enabled grid-less planar routing is favorable for optical layer due to the unique properties of photonics, therefore power consumption driven optical routing can be formulated as a CAD optimization problem under insertion loss (waveguide bending, coupling) and detection constraints (data conversion on photodetectors), etc.

The overall CAD flow for *optical netlist mapping/routing* is illustrated by Fig. 3.14, where a timing-driven procedure is employed to select a proper set of global nets from each core to be routed on the photonic layer. An optimized mapping procedure can result in timing enhancement for the chip since signal

propagation delay on optical interconnect greatly outperforms that of copper interconnect as technology further scales down [22, 86].

After the mapping follows performance (power/timing) driven interconnect routing procedures on both electrical and photonic layer simultaneously. Nets mapped onto photonic layer are routed in a *Gridless Single Layer with Coupling* manner, while the rest of the nets are routed on metal layers. please refer to [44] for details regarding OIL's application to optical routing.

Core-to-Core Network/Packet Routing

Core to core high throughput communication for chip multi-processors have been recently leveraged onto photonic layer for performance ratio enhancement towards Tera-flop super computing scheme [108, 109]. Shown in Fig. 3.12(a) is a traditional many-core-on-chip processor on electrical layer; Fig. 3.12(b) is an photonic network architecture on optical layer for core-to-core communications for Fig. 3.12(a). Major nanophotonic components for constructing (b) are shown in Fig. 3.15, which are drawn in scale with a $3mm$ by 3 mm core. Several current designs of photonic network router [19, 108, 113] R are depicted in Fig. 3.16, where (a)-(c) are relatively large ($\sim$ 500 um) in footprint as non-blocking 4X4 router and (d) is compact in footprint ($\sim$ 70 um) as a blocking 4X4 router. High speed optical waveguide is illustrated as thick lines in Fig. 3.12(b) for mesh-based networks and dotted thin lines will also be adopted for torus based networks. In this work, we use the architecture proposed by [108, 109] for photonic layer core-to-core communication, as part of our *holistic photonic NoC*.

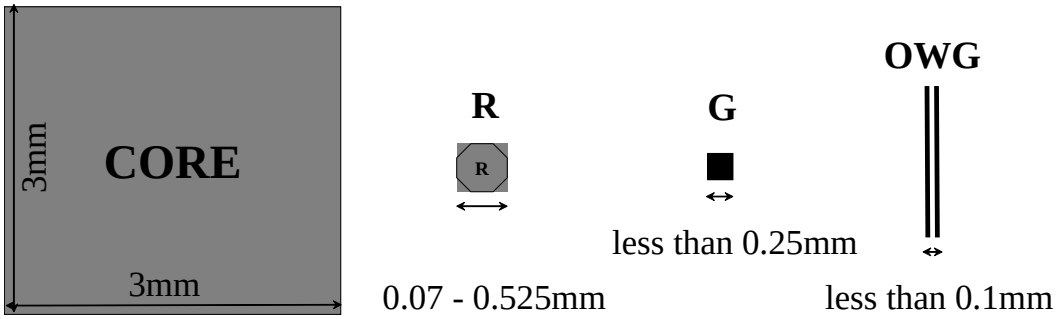


Figure 3.15: Major components for photonic Networks-on-Chip, where CORE is the electrical processor, R is the photonic layer network router, G is the gateway connecting electrical layer and photonic silicon layer, OWG stands for optical waveguide (components drawn in scale based on [19, 108, 113])

3.1.4 Evaluation and Discussion

In this subsection, we will apply OIL to our proposed photonic architecture for some performance evaluations and CAD optimization explorations.

3.1.4.1 Performance Improvement Analysis

Fig. 3.17 illustrates a qualitative perspective of on-chip processing unit (core) performance improvement with the proposed CAD flow in Fig. 3.14, compared with pure electrical interconnect planning. On one hand, the low latency property of photonic interconnect contributes to timing improvement (higher clock frequency) of each core-on-chip; on the other hand, there is also modulation and demodulation time over-head (decreasing as device fabrication technology advances) for data to be converted to and from the optical layer. Under such a scenario, the optimal timing improvement (as marked in Fig. 3.17) corresponds to a proper mapping of a subset of electrical netlist

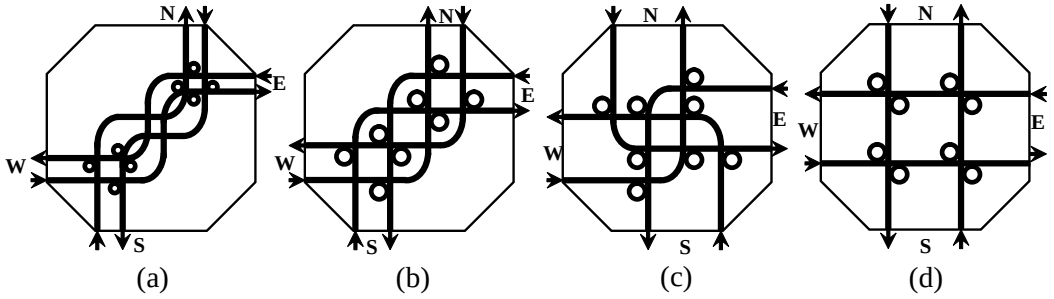


Figure 3.16: Four optical routers for photonic Network-on-Chip from [19, 108, 113],where (a)-(c) are non-blocking photonic network routers and (d) is a blocking photonic network router

from the processing core metal layers onto the photonic layer, which can be formulated as a CAD optimization problem under various constraints, such as power budget, optical interconnect insertion loss, photodetection threshold and integration density on the photonic layer, etc. Applying OIL, we can explore various trade-offs and CAD optimizations for on-chip photonic integration towards next generation high performance chip multi-processor.

3.1.4.2 Interconnect Insertion Loss Analysis

Insertion loss (power loss) is defined as follows in unit of dB, where $Power_{in}$ is the input photon power and $Power_{out}$ is the output photon power of a certain device:

$$dB = -10 \log_{10} \frac{Power_{out}}{Power_{in}}$$

Using OIL, photonic network-on-chip routers in Fig. 3.16 are analyzed as basic building blocks for core-to-core optical routing networks. Best/worst/average losses (coupling loss, waveguide crossing loss and waveguide propagation loss

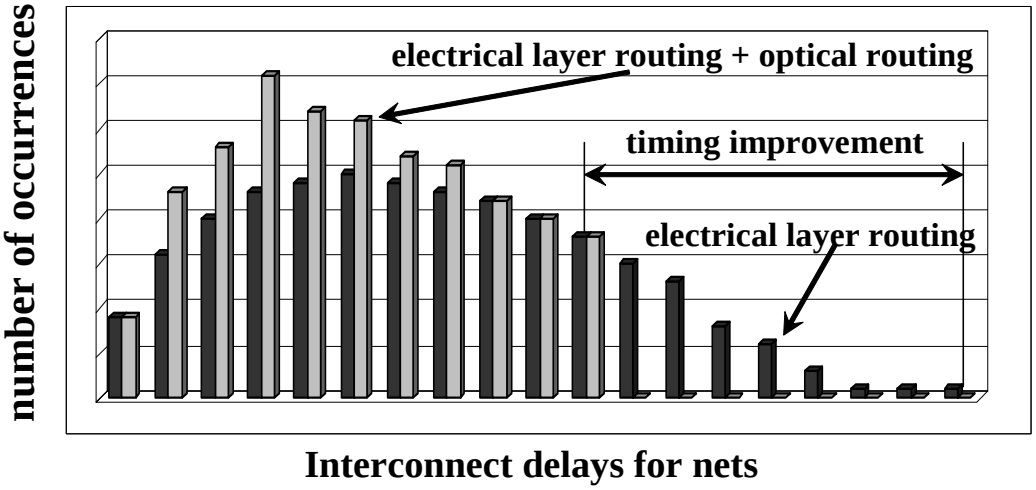


Figure 3.17: Post-routing interconnect delay comparison between the electrical routing and the proposed hybrid routing

considered) for each router are simulated and reported in Fig. 3.18, from which we can see that Router(a) is the most lossy among all three non-blocking routers, while blocking routers such as Router(d) usually has small loss figure due to its simple design structures (less waveguide bending and couplings).

Based on these data, various architecture level analysis can be carried out, such as insertion loss distribution analysis for network packet routing paths to detect/test/validate possible optical-to-electrical data conversion failures at certain *gateways* on a chip, given a specific global network architecture.

3.1.4.3 Multi-Core Scalability Discussion

One of the major challenges for multi-core on chip scaling lies in on-chip memory access bandwidth. As illustrated in Fig. 3.19 curve A, chip

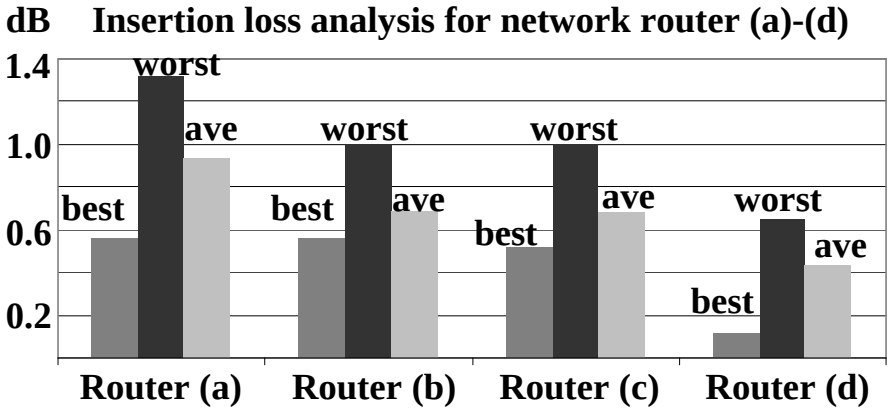


Figure 3.18: Insertion loss analysis for the network routers in Fig. 3.16

performance is expected to decrease as more and more cores are integrated on-chip, with the saturation of on-chip memory access bandwidth resource (curve C). Promising solutions are demonstrated lately, such as 3D IC [77] and RAM-aware NoC routing methodology [63]. Particularly, in the Circa chip project [114] targeted at year 2017, dedicated 3D IC on-chip memory layer and on-chip photonic network layer are combined to deliver scalable memory access bandwidth and high throughput optical communication (e.g., achieving curve B and curve D in Fig. 3.19). Over the years, we expect to see more innovations along this line towards the design of the next generation super computing platforms.

Other challenges need to be properly addressed towards viable on-chip photonic integration include but are not restricted to: high performance low power architecture design and CAD synthesis, further advancement of device fabrication and cost reduction for on-chip nanophotonic devices, etc.

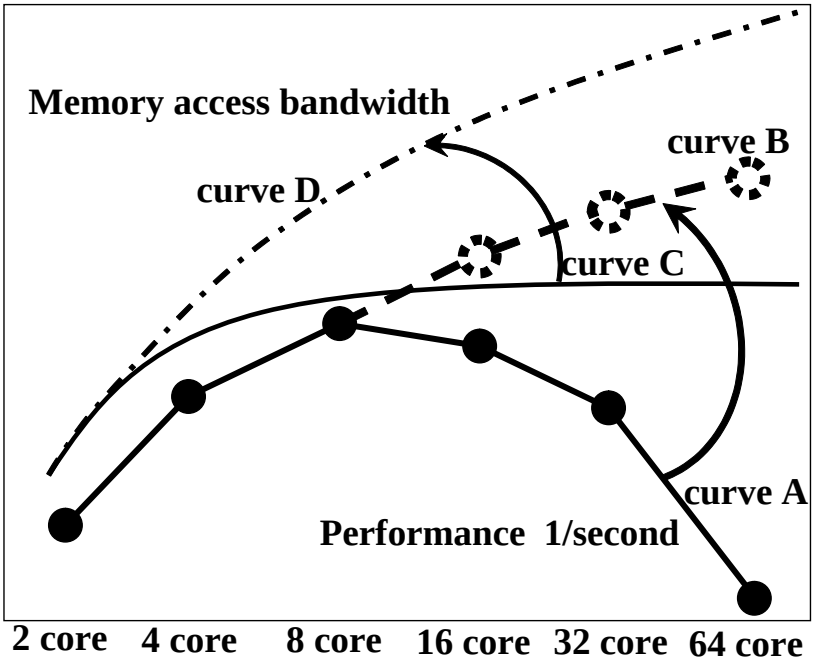


Figure 3.19: Chip multi-processor scalability bottleneck (curve A,C) and potential improvement targets (curve B,D)

3.1.5 Summary

In this section, we proposed OIL (Optical Interconnect Library), a characterized collection of standard cell nano-photonic devices for system level interconnect planning and low power high performance design/synthesis explorations towards a new *holistic photonic Networks-on-Chip* paradigm. Such an architecture incorporates on-chip packet routing (photonic NoC) and within-core wire routing (optical waveguide planning) onto a single optical layer for better photonic silicon integration towards future generation CMPs. The proposed architecture is analyzed and discussed with OIL components for power efficiency, communication latency and potential future work directions.

3.2 Low Power Routing for On-chip Nanophotonic Interconnect Synthesis

As raised in the International Technology Roadmap for Semiconductors [8], silicon system complexity rockets exponentially due to increasing transistor counts, fueled by smaller feature sizes and increasing demands for higher integration / performances with lower costs. Consequently, interconnect design becomes more and more important for DSM VLSI as technology further scales down, among which on-chip optical interconnect is a potential quantum leap towards next-generation technology. Ever since its first introduction by Goodman in [51], the concept of on-chip optical interconnect has attracted more and more attention over the years in industry (e.g., [68, 123]) as well as academia (e.g., [25, 86, 94]), with major focus on device fabrication level. As analyzed and projected in [22], on-chip optical interconnect outperforms traditional copper interconnect in power, throughput and delay with apparent gain below 22nm technology node starting from 2016.

As one of the most promising device level break-through for on-chip optical integration, silicon compatible nano photonic devices (e.g., [56, 123]) take advantage of optical properties of a signal, characterizing great resilience in terms of small delay, low power and high throughput when compared with traditional copper interconnection. Advances in device level improvements of silicon nano photonics (such as photonic crystal structures in [55, 125]) have also been demonstrated. In recent years, low RF power optical modulators operating at a few Gbps speed have been successfully demonstrated [53, 55], with

compact footprint for potential large scale on-chip integration. Compact photodetecters with up to 50Gbps processing rate have also been demonstrated (such as Germanium-on-Insulator photodetecter in [69]). With a proper collection of current Silicon nano-photonic devices and some extended projections / assumptions based on [8, 22], there can be exciting CAD synthesis explorations in interconnect planning for optical on-chip integration.

As a related work, [87, 88] studied timing driven and congestion driven on-chip optical routing CAD algorithms under 3-D system-on-package scenario. Yet the routing geometry in [87, 88] was formulated in a very simple manner: point-to-point straight connection, which also means there is at least 1 optical modulator inserted at each pin and Steiner point in the netlist. There are 3 major issues with such an approach: *First*, it neglects the laser power consumption of optical modulators. Since each modulator requires a laser source for electrical-to-optical data conversion, this approach results in a very power consuming chip; *Second*, it neglects the photon-energy loss constraint on optical interconnect; consequently, there could be pins whose received photon-energy drop below the photo-detectors' detection threshold, leading to inevitable malfunction after optical-to-electrical data conversion. *Third*, optical routing has very different characteristics compared with conventional electrical (copper) interconnect routing, therefore, special routing geometry must be developed to tackle optical interconnect planning problems. In other words, total laser power consumption (proportional to number of modulators inserted) and the constraints for successful optical-to-electrical detection must both be

addressed properly for optimized optical routing geometry.

In this work, we present *O-Router*, an optical routing framework that takes into consideration of various constraints and flexibilities that silicon nano-photonics device libraries and optical waveguide models shall impose on the future on-chip optical interconnect. *O-Router* is driven by low power on-chip silicon nano-photonic integration.

The rest of the section is organized as follows: Subsection 3.2.1 introduces some preliminaries regarding optical and electrical data conversions and silicon photonics, followed by a motivational example and a summary list of key contributions. Subsection 3.2.2 describes our *Optical Interconnect Library* built for *O-Router*; Subsection 3.2.3 focuses on the optical routing Integer Linear Programming (ILP) problem formulation and speed-up techniques, followed by experimental results in Subsection 3.2.4. Subsection 3.2.5 concludes the section with a brief summary and some potential future work.

3.2.1 Preliminaries and Motivations

As shown in Fig. 3.20, on the transmitter's side, the electric signal from the driver (electrical layer) amplitude modulates the light source from the laser inside an optical modulator, and then send the modulated optical signal onto optical interconnect (optical waveguide on optical layer); on the receiver end, a photo-detector detects the photons from the waveguide and converts it into electric signal (back to electrical layer); an amplifier may be needed if this signal drives a high fan-out net on electric layers.

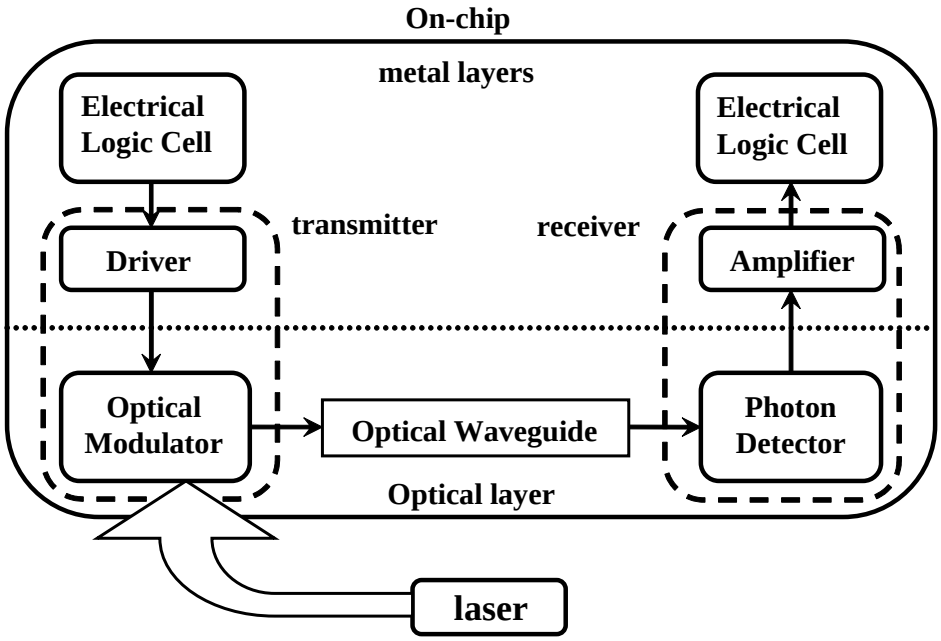


Figure 3.20: Block diagram and working mechanisms for electrical-to-optical and optical-to-electrical data conversions

3.2.1.1 Optical Routing Basics

As aforementioned, optical routing has unique characteristics when compared with traditional copper routing. Manhattan (X/Y) routing based algorithms are not favored on optical layer because of the huge amount of loss caused by the sharp wire turnings along the data path, unless some special structures be inserted; yet these structures are usually costly in fabrication and/or bulky in footprint size, etc.

O-Router performs gridless optical routing with waveguide couplings and crossings on a single layer. As a result, routing geometry becomes very flexible, with different geometries and penalties according to their respective

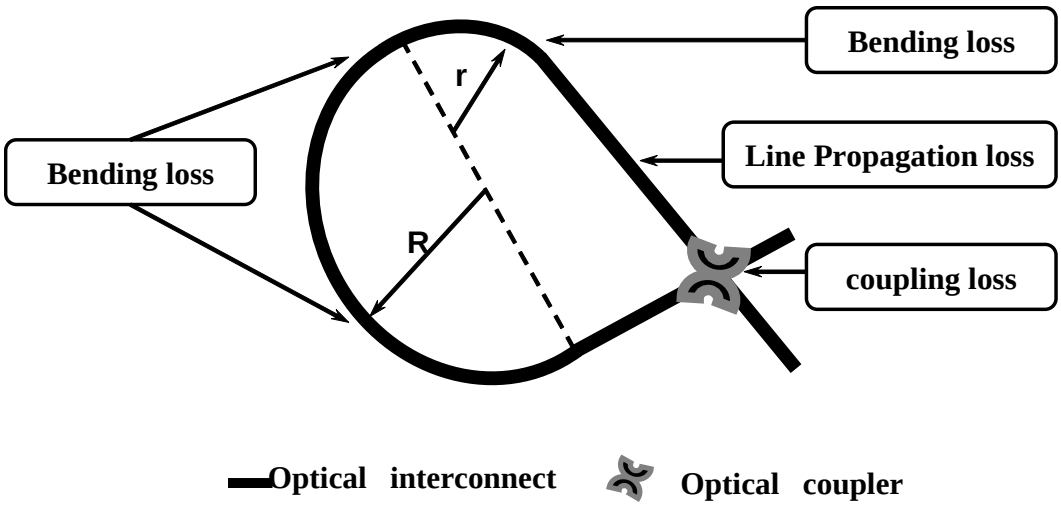


Figure 3.21: Sources of loss for on-chip optical routing

optical interconnect loss. In order to further explore optical routing geometry, we define the following 3 types of losses (with dB unit) on an optical interconnect path in equations 3.5- 3.9.

$$L_{loss} = \alpha \cdot length_{path} \tag{3.5}$$

$$B_{loss} = \beta \cdot \theta \cdot r^{-\eta} \tag{3.6}$$

$$C_{loss} = \gamma \cdot Num_{couplers} \tag{3.7}$$

$$P_{loss} = L_{loss} + B_{loss} \tag{3.8}$$

$$Total_{loss} = P_{loss} + C_{loss} \tag{3.9}$$

As shown in Fig. 3.21, L_{loss} is straight line waveguide loss, it is proportional to the length of optical interconnect, with a coefficient α; B_{loss} is the bending loss, since waveguide cross-section width is negligible compared

to the bending radius in *O-Router*, we assume B_{loss} to be proportional to the degree of the optical interconnect (silicon waveguide) arc angle θ, and inversely proportional to the radius r of the interconnect, with an index η; C_{loss} is the coupling loss, proportional to the number of couplers (crossings) on the interconnect, with a coefficient γ. All related coefficients are determined by our *Optical Interconnect Library*, which is built for *O-Router* and will be explained further in Subsection 3.2.2.

3.2.1.2 Motivational Example

In this subsection, we briefly explore the different trade-offs for optical routing. As shown in Figure. 3.22, there are 2 nets to be routed on a chip, noted as *pini-j*, meaning it is the *jth* member of net i; Fig. 3.22(a) and (b) shows two alternatives for conventional routing on electrical layer with buffers and/or metal via inserted to alleviate the timing penalty caused by the long wires across the chip. Buffers are inserted since RC delay increases quadratically with electrical wire length. Yet buffer insertion is not all-powerful technique. Generally speaking, cross-chip timing critical nets are tough to fix thus impose great difficulty to VLSI design timing closure. As technology further scales down and system integration level rockets, issues with electrical interconnect will get more severe.

Fig. 3.22(c)-(f) show 4 possible routing geometries for the 2 nets on optical layer, according to our optical routing. Routing geometry (c) requires a total of 2 optical modulators: 1 inserted at P1-1, 1 inserted at P2-1.

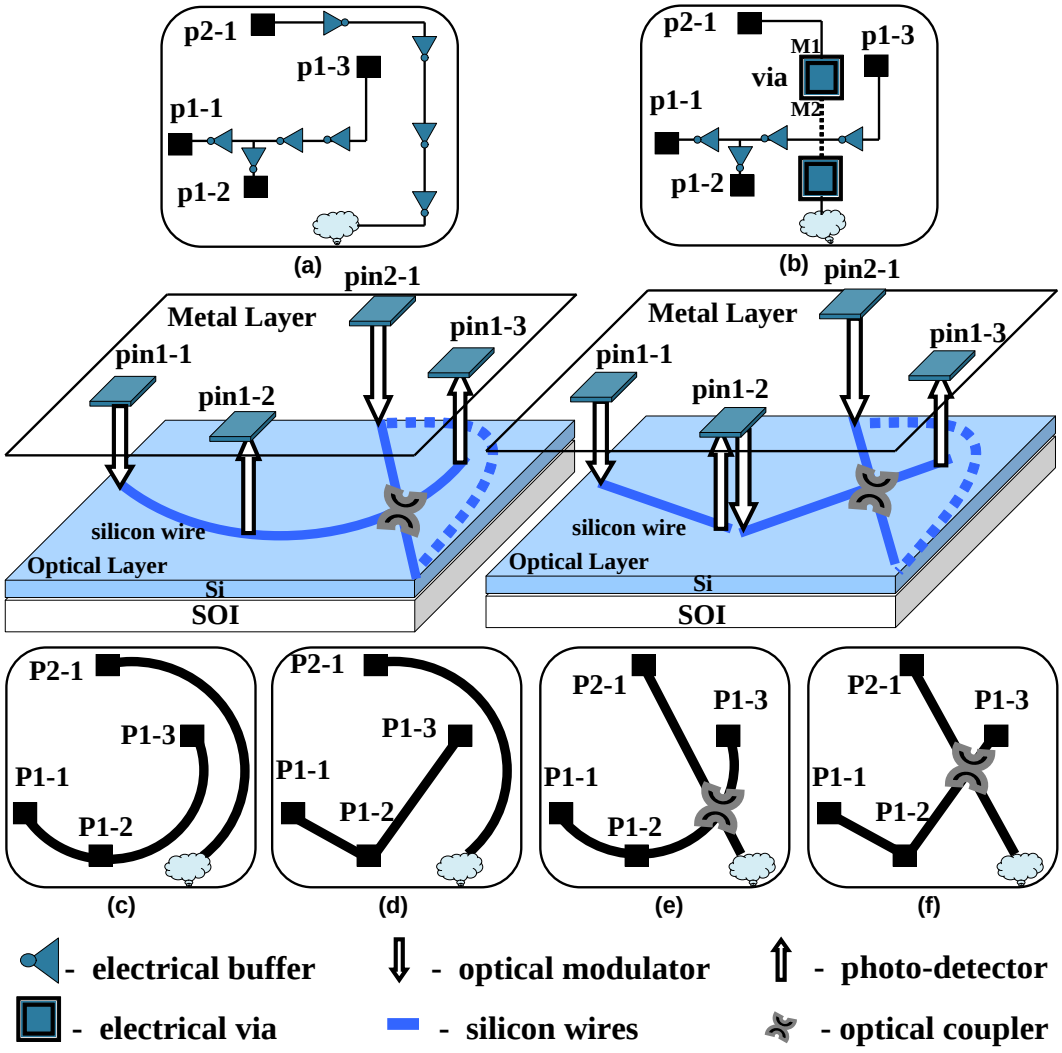


Figure 3.22: Motivational example for electrical routing v.s optical routing

Compared with (c), routing geometry (d) requires 1 extra modulator to be inserted at P1-2 in order to drive P1-3, since sharp turning at P1-2 is either too lossy or too costly to fix other than using an extra modulator. In (e) and (f), optical coupler is introduced for coupling optical signal across 2 wires, with certain amount of loss. In these 2 cases, couplers can be employed either because doing so results in less amount of loss than taking detours as in (c) and (d), or because taking detours results in more coupling loss with other nets on the same chip, etc.

We can learn that geometries (c)(e) result in least among of modulating power among (c)-(f), yet optical interconnect bending loss: B_{loss} is also introduced, as well as the coupling loss: C_{loss} (in (e)) so that the constraint for successful detection at P1-3 may be violated due to too much loss on interconnect. To optimally pick the best routing geometry from the (c)-(f) 4 cases is the motivation of *O-Router*.

O-Router targets at finding optimal optical routing geometry to minimize total modulating power, subject to various constraints imposed by the device characterizations.

3.2.1.3 Main Contributions

Main novelty and contributions of *O-Router* are summarized as follows:

- Based on extensive data collection and road-mapping, we project the technology trends of on-chip silicon nano-photoincs and build *OIL*: an

Optical Interconnect Library characterized for low-power on-chip integration/synthesis.

- For the first time, we formulate the optical routing problem by taking into considerations of various detection constraints and flexibilities that *OIL* imposes on the future optical interconnect.

- Under gridless single layer optical routing with couplings/crossings, the solution space is theoretically infinite. To reduce solution space without losing optimality, we put a set of constraints on the waveguide routing rules and formulate the optical routing problem with Integer Linear Programming.

- We also propose several key techniques to speed-up the optical routing framework under ILP formulation.

Table 3.6: Major OIL components with high level parameters.

	throughput	**length**	**width**	**driving power**	**loss**
modul1	>10Gps	<50um	<10um	1X	-
modulX	>10Gps	<50um	<50um	10X	-
detector	>10Gps	<10um	<10um	-	-
coupler	>10Gps	<50um	<5um	-	<10%

3.2.2 Optical Interconnect Library

To support our *O-Router* framework, we first build an Optical Interconnect Library (OIL), which includes a Mach-Zehnder optical modulator [53], a

photo-detector from [69], a fully simulated optical coupler using Rsoft [6], and a set of optical interconnect (silicon waveguide) model. For details regarding OIL, please refer to [40].

3.2.2.1 Optical Modulator and Photo-detector

Based on existing works [8, 22, 86], we project current OIL parameters towards next generation technology, which essentially enables better on-chip integration for nanophotonic devices.

In Table 3.6, there are 2 sizes of modulators included, one is a normal modulator; the other is ModulatorX: a large modulator with 10X driving power, which will be inserted into a net that suffers greatly from power losses in order to guarantee successful detection. Since the power consumption of ModulatorX is much larger than normal modulator, its usage will be penalized with a constant coefficient $MPow_{penalty}$, details in Subsection 3.2.3.

3.2.2.2 Optical Coupler and Interconnect Model

The working principle of optical coupler is shown by Fig. 3.23. There are 4 ports from A to D for each coupler, and the parallel double interconnect region is the arm region. Optical signals will be cross coupled in the arm region. From the 4 simulation cases, we can verify that PortA=PortD and PortB=PortC always satisfy, as if there is wire connection between A-D and B-C. Optical couplers allow us to make full use of the optical layer routing space, making non-planar netlists routable on a single silicon layer. In case

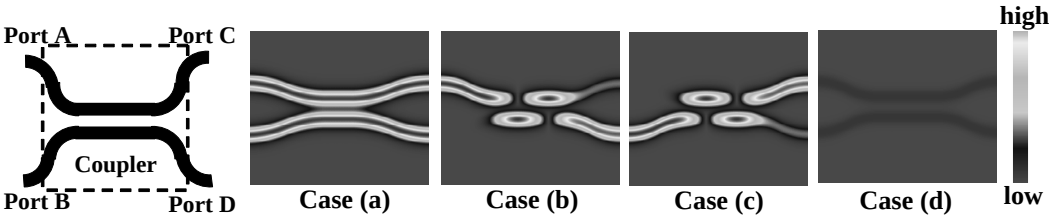


Figure 3.23: Optical coupler in *O-Router* built and simulated in [6]

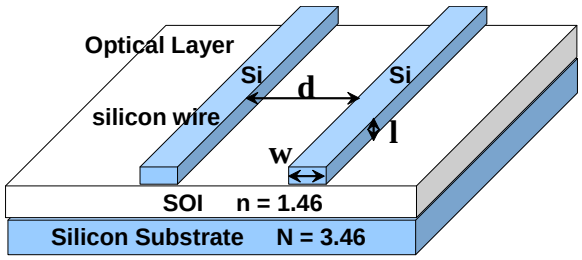


Figure 3.24: OIL on-chip optical waveguide model

(b)(c) in Fig. 3.23, there is slight loss for high optical logic after the coupler, as is formulated by C_{loss}.

As shown in Fig. 3.24, the optical waveguide model included in OIL has a reflective index of 3.46, coated on top of a $2um$ thick SOI layer (reflective index < 1.46). The cross-section width of the silicon wire w=$0.5um$, cross-section height l=$0.22um$, wire spacing d between $0.5um$ and $3.0um$. d should be set properly to avoid wire cross-talk.

3.2.3 O-Router Formulation and Algorithms

Given the pin locations of certain placed netlist for optical routing, *O-Router* seeks optimal routing solution with Integer Linear Programming to minimize total modulating power, meanwhile satisfying various detection con-

straints according to established OIL parameters. This subsection is divided into three parts: First is the optical netlist mapping. This is when suitable optical netlist benchmarks for *O-Router* are constructed. Second part is the core problem formulation of the low power optical routing. The third part is about the routing speed-up techniques.

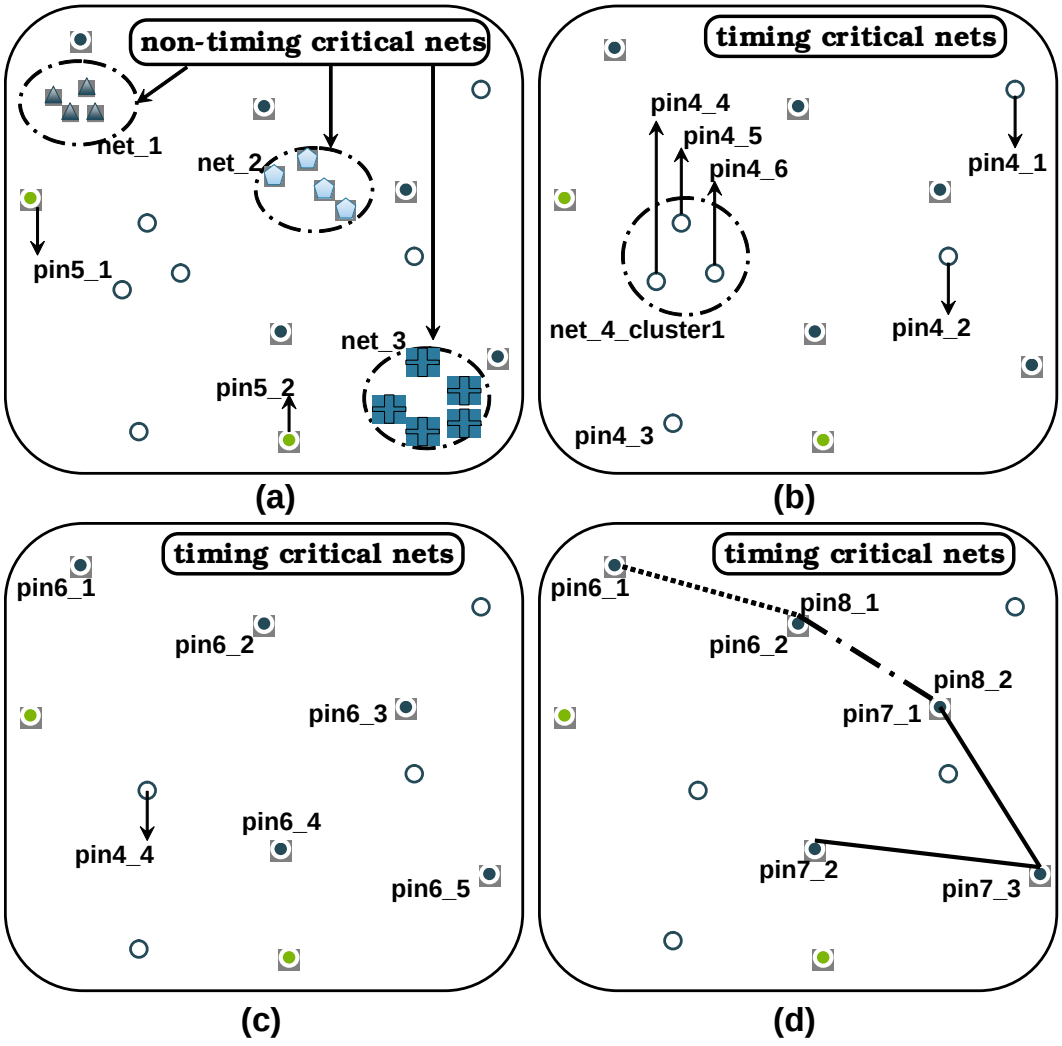


Figure 3.25: Illustrations for *optical netlist mapping*

3.2.3.1 Optical Netlist Mapping

Given an electrical layer netlist after placement, the goal of this step is to prepare an optical netlist that makes most use of optical layer resource to fix top timing critical nets (i.e., longest) in electrical layer. For our ILP formulation, the resulting optical netlist of this stage consists of only 2, 3 and 4 pin nets. It takes place in 3 phases:

Phase 1: Pre-select top timing critical nets from electrical layer to map onto optical layer. Shown in Fig. 3.25(a)(b), non-timing critical nets 1/2/3 are not selected.

Phase 2: Cluster within each pre-selected net for routing efficiency enhancement. Since optical routing is most effective dealing with global interconnect, we map a single pin from each local pin cluster onto the optical layer and leave the remaining pins to electrical layer.

As shown from Fig. 3.25(b) to Fig. 3.25(c), the *net_4_cluster1* is represented by *pin4_4* on optical layer. With this phase, the pin number for each optical net becomes very small. For *O-Router*, we manage to keep each optical net size to below 5 pins. Practically, nets with more than 4 pins can be decomposed into a set of 2/3/4 pin nets, as illustrated in (c)-(d), where a 5-pin net6 is decomposed into two 2-pin nets and a 3-pin net.

Phase 3: For intersected 2-pin nets in the netlist from Phase2, expand them to have 2 more integer variables if and only if they can avoid crossing each other by taking an arc detour, meanwhile the detour does not cut a third

net. This step further expands the feasible solution space for 2-pin nets.

3.2.3.2 Integer Linear Programming Formulation

For the original ILP formulation, we enumerate all routing geometries for the 2-pin, 3-pin and 4-pin nets, shown in Fig. 3.26 (a concave shape 4-pin net is shown as an example). Each X_{ij} is an integer variable, where i $\in$ *net_space*, $j \in$ *sol_space(net i)*. When $X_{ij} = 1$, the corresponding routing geometry from Fig. 3.26 will be adopted, as part of the final routing solution space. Number of modulators in each X_{ij} is also recorded; OIL will return the actual modulating power based on this number and the ij index.

The ILP formulation is as follows in Equation 3.10- 3.19, with all terms and variables explained in Table 3.7. The objective function is the total power required to drive all the on-chip optical modulators for our optical interconnect framework. The ILP solver will minimize the objective function, subject to constraints imposed from Eq. 3.11 to Eq. 3.19. In Eq. 3.10, the first term $MPow_{Xij}$ is total modulating power consumption for routing geometry X_{ij} using 1X modulators, while the second term $(MPow_{penalty} - P_0) \cdot M_{ij} \cdot N_{ij}$ is for penalizing the usage of 10X driving power ModulatorX: if M_{ij} is 1(hard constraint violation), then ModulatorX will be used to replace all Modulator1's in geometry X_{ij} to meet the constraint (P_0 is the laser power consumption of Modulator1).

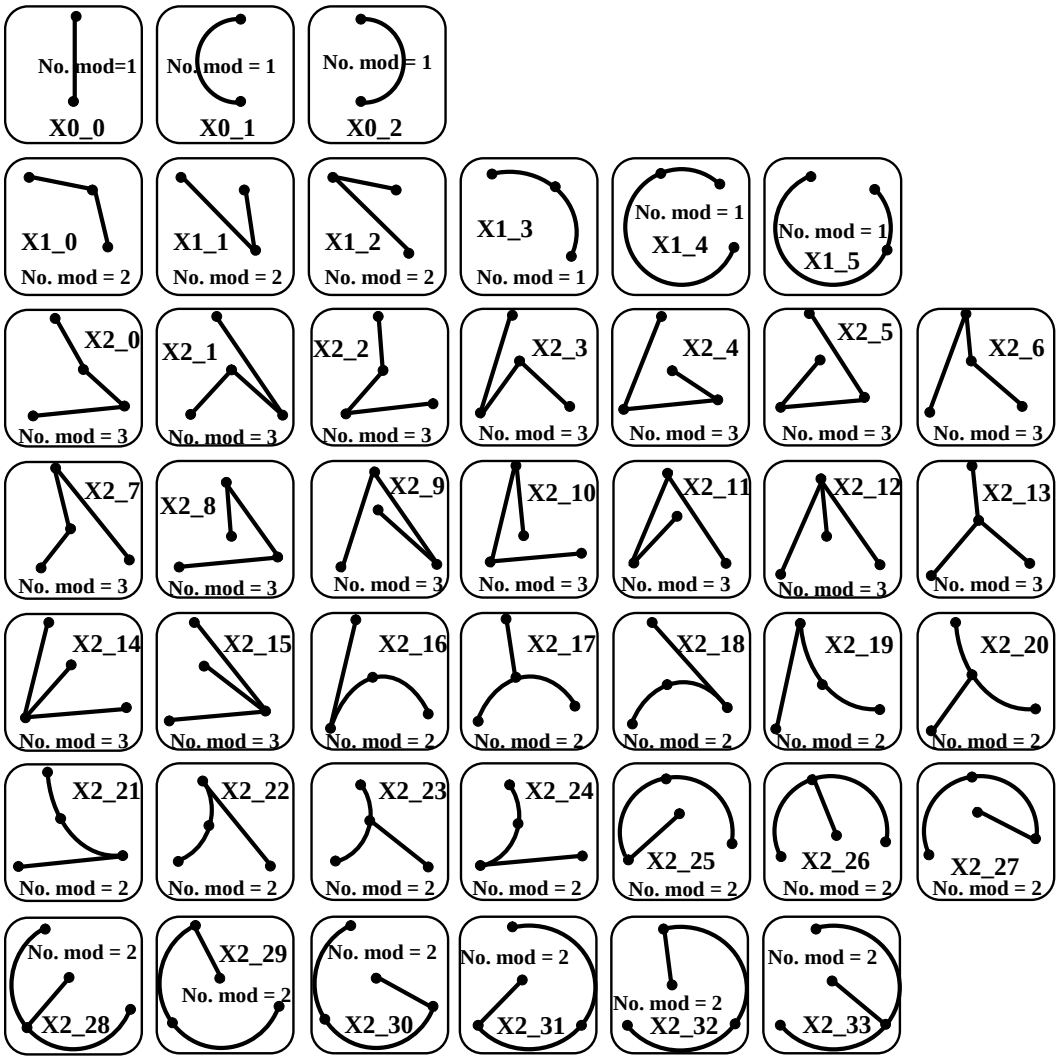


Figure 3.26: A list of optical routing geometries (represented by integer variables) for 2, 3 and 4 pin nets

Table 3.7: Descriptions for ILP involved terms and variables.

Name	Description
$net\_space()$	set of nets for an optical netlist
$sol\_space(i)$	set of possible routing geometries for net i
$MPow_{Xij}$	total modulator power consumption of routing geometry X_{ij}
$MPow_{penalty}$	power consumption penalty for using each ModulatorX. Set to 10 times of P_0
P_0	power consumption of Modulator1
N_{ij}	least number of optical modulators used for geometry X_{ij}
$C_{loss_{Xij}}$	coupling loss power between routing geometry X_{ij} and X_{mn}
$P_{loss_{Xij}}$	propagation loss power on silicon wires of X_{ij}
X_{ij}	integer variable. $X_{ij} = 1$ means to accept the jth routing geometry of net i
M_{ij}	integer variable. $M_{ij} = 1$ means to insert modulatorX into jth routing geometry of net i
X_{ij_mn}	integer variable. numerically equals to $X_{ij} \cdot X_{mn}$
$loss\_th_{Xij}$	loss threshold for O-E conversion for X_{ij}
pow	more driving power each ModulatorX brings than Modulator1
γ_{ij_mn}	coupling loss coefficient returned by OIL dependent on geometry X_{ij} and X_{mn}

$$min\{\sum_{i \in net\_space()}^{j \in sol\_space(i)} [MPow_{Xij} \cdot X_{ij} + (MPow_{penalty} - P_0) \cdot M_{ij} \cdot N_{ij}]\} \quad (3.10)$$

$$s.t$$

$$\forall i, m \in net\_space, i \neq m, j \in sol\_space(i), n \in sol\_space(m) :$$

$$P_{loss_{Xij}} \cdot X_{ij} + net_{loss_{Xij}} \leq loss\_th_{Xij} + pow \cdot N_{ij} \cdot M_{ij} \quad (3.11)$$

$$P_{loss_{Xij}} = L_{loss_{Xij}} + B_{loss_{Xij}} \tag{3.12}$$

$$net\_loss_{Xij} = \sum_{\substack{m \in net\_space \\ n \in sol\_space(m)}} C_{loss_{Xij\_mn}} \cdot X_{ij\_mn} \tag{3.13}$$

$$C_{loss_{Xij\_mn}} = \gamma_{ij\_mn} \cdot cross\_num < X_{ij}, X_{mn} > \tag{3.14}$$

$$X_{ij} + X_{mn} \leq 1 + X_{ij\_mn} \tag{3.15}$$

$$(1 - X_{ij}) + (1 - X_{mn}) \leq 2 - 2X_{ij\_mn} \tag{3.16}$$

$$\sum_{j \in sol\_space(i)} X_{ij} = 1, \quad where \;\; X_{ij} = 0 \; or \; 1 \tag{3.17}$$

$$X_{ij\_mn} = 0 \; or \; 1 \tag{3.18}$$

$$M_{ij} = 0 \; or \; 1 \tag{3.19}$$

Constraint Eq. 3.11 is set for each routing geometry X_{ij}, such that its total loss (propagation loss P_{loss} and coupling loss C_{loss}) is bounded by an upper bound of loss threshold: $loss\_th_{Xij}$, once the upper bound of loss is exceeded, it means the photo-detection requirements in routing geometry X_{ij} are violated. If among all feasible X_{ij}, some of such constraint is inevitably violated, then ModulatorX will be inserted into the corresponding geometry X_{ij} and replace existing 1X modulators. Constraint Eq. 3.14 explicitly maps the crossing number of a net into corresponding coupling loss using OIL.

For ILP formulation of the calculation of optical interconnect coupling number, we introduced the cross-term integer variables: X_{ij_mn}. Numerically, it is the product of term X_{ij} and X_{mn}. Since variable multiplications are not supported by ILP solver, we add the constraint pair Eq. 3.15- Eq. 3.16. Integer constraints Eq. 3.15 and Eq. 3.16 bound the X_{ij_mn} term so that it always

Algorithm 12 ILP based Optical Routing for low power chip

Require: mapped optical netlist benchmark
  invoke optical netlist parser; **link** OIL
  while $i \in net\_space$ **do**
    while $j \in sol\_space(i)$ **do**
      calculate ($L_{loss_{Xij}}$,$B_{loss_{Xij}}$,$MPow_{Xij}$,$MPow_{penalty}$,etc.)
      while $m \in net\_space, m \neq i$ **do**
        while $n \in sol\_space(m)$ **do**
          calculate ($C_{loss_{Xij\_mn}}$, constraint coefficients,etc.)
        end while
      end while
    end while
  end while
  generate glpk syntax file; **invoke** glpk ILP solver – minimize
  return optical routing for minimum modulating power

equals the product of its two corresponding routing geometries. Equality constraint Eq. 3.17 makes sure that the ILP solver eventually picks only 1 routing geometry out of each net for the final optimal solution. For further details please see Table 3.7 and Algorithm 12.

3.2.3.3 Variable Reduction and Speed-up Techniques

A direct implementation of Algorithm 12 will result in very large number of variables and huge amount of computation, especially for large optical netlists. We propose techniques to speed-up *O-Router*.

Variable Trimming/Merging

Variable trimming procedure first scans through the X_{ij} list and calculate bending loss B_{loss} and line propagation loss L_{loss} for each X_{ij}. If the loss of X_{ij} itself becomes unbearable, then such a routing geometry is dumped

Algorithm 13 ILP variable number reduction via trimming

Require: mapped optical netlist benchmark
  while $i \in net\_space$ **do**
    while $j \in sol\_space(i)$ **do**
      calculate ($L_{loss_{Xij}}$,$B_{loss_{Xij}}$)
      if $L_{loss_{Xij}} + B_{loss_{Xij}} \geq threshold_{Xij}$ **then**
        exclude X_{ij}; update data structures
      end if
    end while
  end while
  return trimmed set of routing geometries for each net

before invoking ILP. Variable trimming procedure successfully trims off the infeasible integer variables in Fig. 3.26 and greatly reduces the variable set. Solution optimality will not be harmed with careful choice of the *threshold* value. Details about this procedure are shown Algorithm 13.

Variable merging procedure runs in parallel with *variable trimming procedure*. As described in Algorithm 14, it cuts the cross-variable set to half of original size with a simple observation:

$$X_{ij} \cdot X_{mn} = X_{ij\_mn} \tag{3.20}$$

$$X_{mn} \cdot X_{ij} = X_{mn\_ij} \tag{3.21}$$

$$X_{ij\_mn} = X_{mn\_ij} \tag{3.22}$$

Essentially, X_{ij} and X_{mn} generate non-zero constraint coefficients only when both of them are adopted, which means the product equality holds in Eq. 3.20 and 3.21, consequently, Eq. 3.22 also holds, so we can rename half

Algorithm 14 ILP cross-term variable reduction via merging

Require: mapped optical netlist benchmark
  while $i \in net\_space$ **do**
    while $j \in sol\_space(i)$ **do**
      while $m \in net\_space, m \neq i$ **do**
        while $n \in sol\_space(m)$ **do**
          if $i > m$ **then**
            swap (i,j) with (m,n) in X_{ij_mn}; **calculate** cross-term constraint coefficients; **update** glpk syntax file
          end if
        end while
      end while
    end while
  end while
  return reduced set of cross-terms

of the cross-term variables to the other half, since they are identical. Details about this procedure are shown in Algorithm 14.

Bounding Box Elimination for Speed-up

The introduction of Bounding Box contributes to computation speed-up of O-Router. *Bounding Box* of net i is defined as the rectangle that bounds all the pins of net i. It is defined by 4 values as in Eq. 3.23:

$$Bounding\_Box_i = (min(X), max(X), min(Y), max(Y)) \tag{3.23}$$

$$where \quad X \in x\_axis\{net_i\}, \quad Y \in y\_axis\{net_i\} \tag{3.24}$$

As illustrated in Fig. 3.27, a *Bounding_Box_Matrix* will be generated in pre-scanning stage; for any pair of nets with non-overlapping bounding boxes, a 0 is recorded, otherwise, 1 is written; In Fig. 3.27, *net_j* and *net_k*

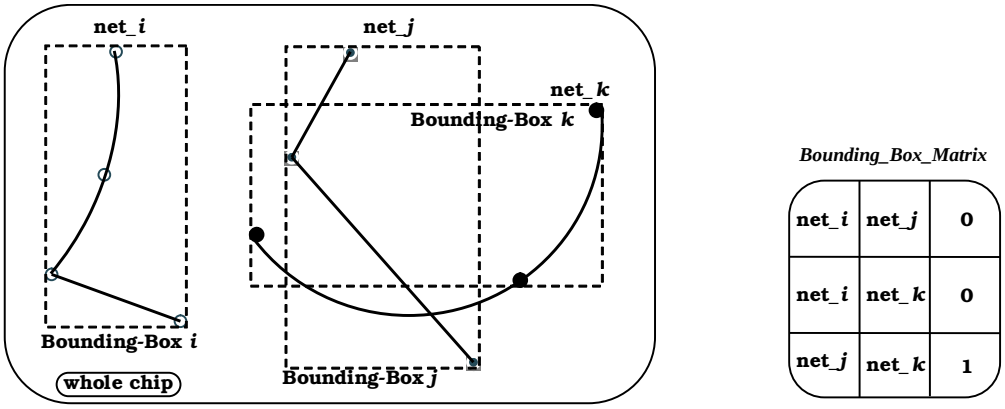


Figure 3.27: Illustration of Bounding Box Elimination

have potential crossings, thus only them will be processed for constructing coupling loss constraints. This procedure worth the efforts because a general algorithm for calculating C_{loss} is much more complicated than min/max value search. Further details for bounding box elimination procedure are shown in Algorithm 15.

With all 3 speed-up procedures, the original ILP formulation is modified, implemented and tested.

Algorithm 15 bounding box for C_{loss} computation speed-up

Require: mapped optical netlist benchmark
  generate *bounding_box_matrix*[ ][ ]
  while $i \in net\_space$ **do**
    while $m \in net\_space, m \neq i$ **do**
      if $bounding\_box\_matrix[i, m] == 1$ **then**
        calculate $C_{loss_{ij\_mn}}$; update glpk syntax file
      end if
    end while
  end while
  return optical routing for minimum modulating power

3.2.4 Experimental Results

Simulations are carried out according to the aforementioned 3 steps in Subsection 3.2.3, and original electrical benchmarks come from ISPD98/08 routing benchmarks. ibm01-04 are the final 4 optical netlists benchmarks, listed as in Table 3.8. Due to considerations of silicon wire spacing/low coupling noise communication, the sizes of the optical netlists are kept from small to medium, and the optical layer pin density is kept from low to medium. As a baseline for *O-Router*, Minimum Spanning Tree (MST) routing algorithm is implemented on ibm01-04. Both *O-Router* framework and MST algorithm are repeated on ibm01-04 for 2 different photo-detection threshold values: 55% and 75%. Such percentages signify the photo-detection power threshold for received signals at the end of optical interconnect. Therefore, 75% threshold photo-detectors impose stricter detection requirements on *O-Router* framework. In Table 3.8, the simulated power consumptions are normalized by the amount of power reported by *O-Router* on ibm01, under photo-detection threshold of 55%. For 55% threshold, *O-Router* achieves above 50% of power reduction compared to MST baseline, with a max of 81.1% on ibm04. For the 75% threshold, *O-Router* reports slightly less power reductions due to higher detection requirements; still an average of above 50% reduction, with a max of 67.3% of power reduction on ibm04.

Table 3.8: Performance comparisons between *O-Router* and Minimum Spanning Tree algorithm.

	photo-detection threshold : 55%				photo-detection threshold: 75%			
	ibm01	ibm02	ibm03	ibm04	ibm01	ibm02	ibm03	ibm04
Net number	5	20	50	137	5	20	50	137
Pin number	15	50	155	391	15	50	155	391
Pin/net ratio	3	2.5	3.1	2.85	3	2.5	3.1	2.85
MST-routing (normalized power)	3.5	6	35.66	305.13	3.5	12.75	39	306.25
O-Router (normalized power)	1	2.88	10.75	57.75	2.13	5.38	16.5	100.25
Improvement	71.40%	52.00%	69.90%	81.10%	39.10%	57.80%	57.70%	67.30%

3.2.5 Summary

In this section, we presented an optical routing framework, *O-Router* for low power synthesis of on-chip silicon nanophotonics with consideration of various detection constraints. Based on ILP formulation with several variable reduction techniques for routing speed-ups, *O-Router* utilizes Optical Interconnect Library, which is an established collection of some silicon compatible on-chip nanophotonics devices and optical interconnect models, with key parameters projected for future technologies based on optical interconnect roadmap. Experimental results show promising improvements compared with traditional Minimum Spanning Tree routing algorithm. We expect to see a lot of future works along this direction as new nanophotonics devices are introduced for the ultimate global optical and electrical interconnect co-synthesis and planning.

3.3 Low Power Thermal-Reliable WDM Placement for On-Chip Nanophotonic Interconnect

As semiconductor technology roadmap[8] extends into deeper sub-micron domain, the development of future generation high performance low power silicon systems faces many key challenges. Among them, VLSI interconnect plays more and more critical roles with reasons of manifold: (1) growing interconnect versus gate delay ratio; (2) longer global interconnects due to higher levels of on-die functional integration; (3) tougher timing closure due to higher operating frequencies and more complex chip designs; (4) more challenging interconnect design for power efficient many-core chips.

To address the interconnect challenges for next generation processors, various alternative techniques have been proposed as potential solutions (e.g, [21, 86, 116]). Among them, nanophotonics devices and interconnect attract active researches (e.g., [64, 71, 86, 93, 123, 140]) due to their unique potential for designing high speed and low power on-chip communication links. However, most success in photonics has been limited to the off-chip scale, due to various constraints for on-chip integration, such as large device footprints, high power consumptions and fabrication limitations.

In recent years, various nanophotonics devices were demonstrated with great on-chip integration potential, among which there are micro/nano-scale optical modulators, photo-detectors, couplers, switches, waveguides and WDM (Wavelength Division Multiplexing) devices, etc. Together, they provide rich resources for device modeling/roadmapping [71] and on-chip integration [20,

40, 44, 87, 98, 110] opportunities, meanwhile bringing new perspectives to the traditional architectural and physical design methodologies.

Lately, nanophotonics has been actively applied to on-chip networks and special architecture designs (e.g, [65, 78, 98, 110]) to generate dynamic data traffic routing with high throughput Time Division Multiplexing and WDM on-chip nano-photonic links. These works show promising potential for employing nanophotonics to address inter-core and memory bandwidth limitations of future processor chips. However, thermal reliability issue has not been considered properly in currently existing photonic architecture/network designs. In reality, such designs may totally fail to function since nanophotonic devices are prone to on-chip temperature variation, especially for ring resonator structures which are widely used for compact WDM applications.

Meanwhile, studies for nanophotonics CAD physical design have been very limited. An early work was presented by [87, 88], where straight-line single-channel optical waveguides were employed for a system-on-package optical routing framework under timing driven metrics. However, physical device characterizations were not considered for modulators, waveguide, photodetectors, and important issues such as optical link configuration, loss figure, thermal reliability and signal integrity were not properly explored. In [20], physical-layer effects (loss, power) are considered and applied to photonics Network-on-Chip performance evaluation. Yet without the support of a proper CAD environment, it is very difficult to design photonics architectures with optimal performance meanwhile pushing to the edge of power budget. In [40, 44],

a parameterized nanophotonics interconnect library was presented together with a CAD physical design framework for low power on-chip optical routing, indicating great on-chip integration potential via proper architectural/physical co-design and CAD flows for generic nanophotonic link construction. Unfortunately, thermal reliability models are not included and WDM mechanisms are not utilized.

In this section, we investigate the current pool of nano-photonics devices and further explore their on-chip integration potentials by presenting *GLOW*: a global routing framework for low power thermal-reliable on-chip WDM interconnect design and synthesis. The rest of the section is organized as follows: in Subsection 3.3.1, we motivate the WDM based optical routing problem under the critical consideration of thermal reliability and summarize the main contributions. In Subsection 3.3.2 we extend [40] by introducing thermal and power related models of various nanophotonics devices. Different types of on-chip optical links are also analyzed in terms of timing, thermal and power. In Subsection 3.3.3 we present an overview of our proposed CAD flow, followed by Subsection 3.3.4 and 3.3.5, in which we explain the detailed formulation and algorithms for *GLOW* together with an alternative heuristic approach *CAT*. Subsection 3.3.6 presents and discusses the simulation results, followed by conclusion in Subsection 3.3.7.

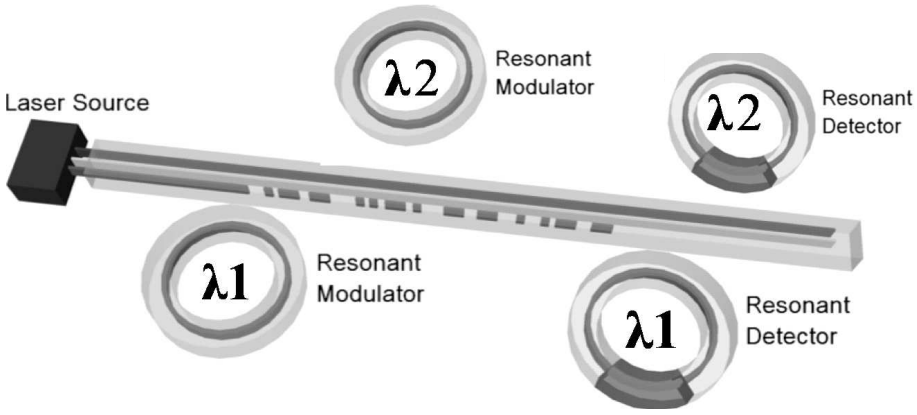


Figure 3.28: A data link with on-chip nanophotonic WDM [98]

3.3.1 Motivation and Contributions

With on-chip WDM mechanism (Fig. 3.28) providing great signal multiplexing capacity and power efficiency, we motivate a global routing strategy to take the advantages of WDM channel bands under various physical design constraints such as thermal reliability and timing, etc. Before going to further details, we introduce the following definitions for the convenience of discussion.

Definition 3.3.1. WDM *link*: A piece of on-chip interconnect that solely or partially employs wavelength division multiplexing mechanism. A WDM link is consisted of laser source, on-chip nanophotonic waveguide (OWG) and modulation/detection devices, etc.

Definition 3.3.2. WDM *trunk*: The body of the OWG in a WDM link is also referred to as a WDM trunk.

Definition 3.3.3. WDM *channel*: The working bands (wavelengths) of a

WDM link are defined as channels. Each channel of a WDM link is assigned a unique wavelength λ which signifies the carrier frequency of the optical signal.

We first briefly describe a motivational example with a very simple scenario as illustrated in Fig. 3.29. Given 2 nets (A,B,C,D)(X,Y,Z) to be routed with node A and X as the drivers, B,C,D,Y,Z as sinks, we aim to find a global routing solution in the optical-electrical domain to satisfy:

- Thermal reliability and functionality
- Minimal driving power required
- Signal integrity and data conversion quality
- Timing considerations
- WDM channel utilization rate
- Synthesis legalization based on opto-electrical domain design rules

In Fig. 3.29, thermal issue refers to the case when on-chip temperature variation causes extra power loss, signal degradation or even malfunction to the optical modulators, photo-detectors and WDM waveguide. Without careful considerations and planning, an opto-electrical link could fail to operate in reality due to fallacies introduced by high thermal variations. In the extreme cases, we can simply set the thermal killer regions as blockages in the routing stage, yet still moderate temperature variation affects the rest of the chip that suffers from different degrees of thermal induced power loss. Under power objective, this power loss must be minimized although it would not cause circuit malfunction. Power loss also comes from waveguide crossings and bending.

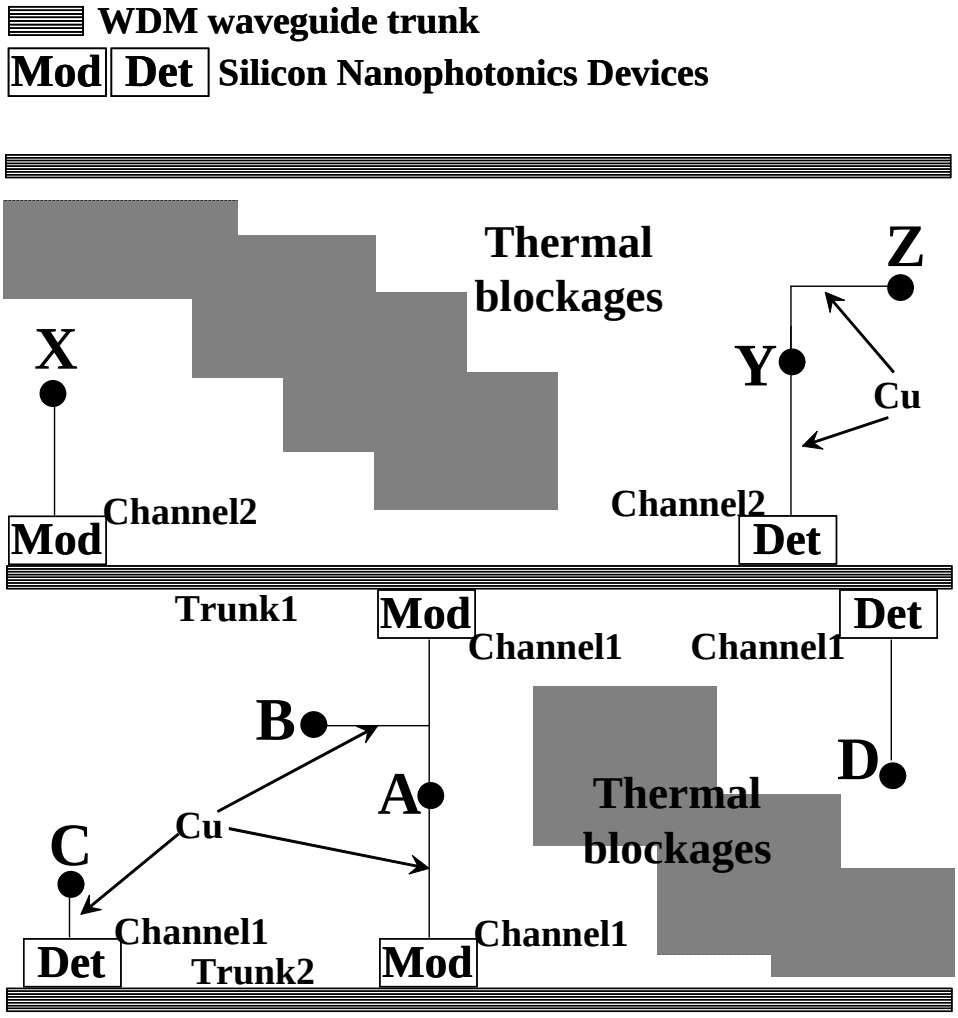


Figure 3.29: A motivational example for thermal-aware optical routing featuring on-chip WDM

During routing in the opto-electrical domain, timing condition must be met such that a hybrid data link does not generate longer signal delay than an otherwise routing path in the electrical domain. Under such conditions in Fig. 3.29, link A→B is routed with Cu interconnect while links A→C, A→D are partially merged with WDM trunks, meanwhile link A→D takes trunk1 due to the thermal blockage between sink D and trunk2. Similarly, link X→Y takes path through trunk1 due to the thermal blockages on the top and right side of X.

For high WDM channel utilization rate, sharing onto a single WDM trunk is encouraged unless timing and/or thermal conditions are violated. In this case, path A→C would tend to merge with link A→D onto trunk1, however is prohibited by the long delay from trunk1 to sink C.

As depicted by Fig. 3.29, WDM trunk1 has 2 WDM channels assigned: Channel1 with link A→D, Channel2 with link X→Y. Trunk2 only has 1 channel assigned: Channel1 with link A→C. In principle, the more channels utilized onto one single WDM trunk, the less trunks totally need to be fabricated. However, in cases when on-chip thermal variations get complicated, a proper trade-off need to be sought between total power consumption and total number of trunks.

Last but not the least, the final routing solution needs to deliver signals strong enough to be picked up by the photo-detector, meanwhile be legalized according to design rules in both optical and electrical domains. Most of the latter can be done in the detailed routing stages.

We summarize the key contributions of this section as follows:

- We perform nanophotonics device characterization and modeling
- We propose a CAD framework for on-chip WDM synthesis to co-optimize power and thermal reliability
- We develop thermal reliability models of various nanophotonics devices
- We formulate the global routing problem with Integer Linear Programming techniques
- we evaluate the CAD framework with various testcases derived from ISPD global routing benchmarks

3.3.2 Nanophotonics Device Models

Quantified design space exploration and CAD optimization require a properly selected and well parameterized set of nano-photonics elements/devices to build high performance on-chip optical links that are power efficient and thermal reliable. Therefore, we extend [40] with WDM related modules and thermal models to configure/analyze different on-chip optical links, with respect to critical considerations in the physical design stages, such as power, loss, timing, temperature variation and thermal-reliability.

As an extensible open source of device models, current OIL includes on-chip optical modulators, photodetectors, buffers, switches, couplers, optical waveguide and on-chip WDM devices, etc. As depicted in Fig. 3.30, OIL

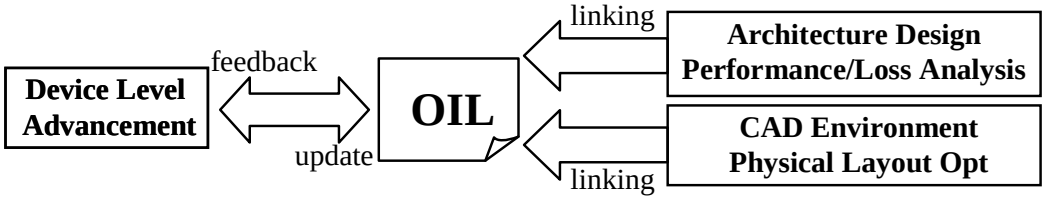


Figure 3.30: OIL (Optical Interconnect Library) modeling serving as crux in-between of device/CAD/architecture design

enables closer interaction between device fabrication, architecture design and CAD optimization for a promising nano-photonics interconnect solution. For OIL release and updates, please refer to [7].

3.3.2.1 Device Characterization

An *optical link* is designed with a combination of certain modulator, OWG (on-chip optical waveguide), photo-detector and corresponding driver (amplifier) circuits, as shown in comparison to a Cu wire in Fig. 3.31. Various types existing nanophotonic devices enables us to configure different high performance optical links in terms of power and/or speed. Based on current photonics fabrication technology, optical signaling has great advantage over low-K Cu interconnect (11ps versus 37ps per mm on Metal5/6).

With careful CAD design methodologies, cross-domain opto-electrical interconnect synthesis harbors great potential for both performance(timing) improvement and power reduction. In particular, we use optical modulators to convert electrical signals into optical (E-to-O) domain (onto OWG channels), and photo-detectors to convert the light pulses into electrical (O-to-E) domain

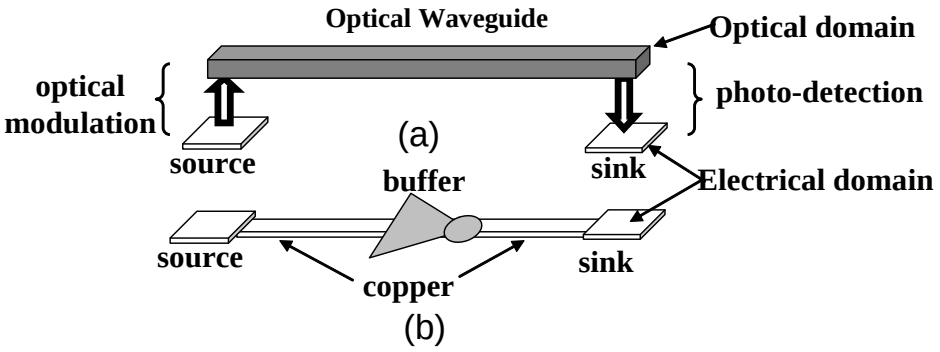


Figure 3.31: A data link comparison between optical and Cu interconnect

under detection constraints. Couplers are also employed to enable optical waveguide cross-couplings for planar routing. These required building blocks of on-chip optical links are characterized with respect to device operating speed, optical/electrical power consumption, on-chip loss and footprint, etc. Table 3.9 summarizes the devices used.

From Table 3.9, we can draw several observations as follows, First, multiple devices as modeled enable us to configure different on-chip optical links featuring low power, high performance or a trade-off in-between. Second, signal propagation speed is estimated to be 3.4X (11ps v.s 37ps) faster on currently fabricated OWG than a global Cu interconnect under optimal repeater insertion in $22nm$ technology. However, further studies must take into considerations the modulation and detection delay for E-to-O and O-to-E data conversions. Also, each sink's photo-detection optical power threshold must be satisfied for successful O-to-E conversion, which is 100μW in Table 3.9. Last but not least, photo-detection speed should be lower bounded by modulation

Table 3.9: Device and interconnect model details

	footprint	speed	on-chip loss	E-power
mod_M1	100X10um	1.6Gb/s [28]	5dB	0.86mW
mod_M2	1000X40um	30Gb/s [64]	5dB	0.5W
mod_R1	30X40um	3.0Gb/s [24]	2dB	0.26mW
mod_R2	30X40um	14Gb/s [46]	2dB	0.7mW
	footprint	speed	O-det power	E-power
det1	10X10um	5Gb/s	0.1mW	0.17mW
det2	20X20um	40Gb/s	0.1mW	1.3mW
	radius	coupling gap	pass loss	couple loss
coupler	1-2um	0.2um	0.01dB	0.5dB
	delay	optical loss	thickness	width
WDM	11ps/mm [71]	1.5dB/cm	230nm	450nm
	delay	thickness	width	repeater
Cu	37ps/mm	1um	0.4um	per 1.4mm

Cu interconnect on $22nm$ technology Metal5/6 with ρ=2.2$\mu\Omega$·cm, R_{sheet}=0.022Ω, C=2pF/cm. MOSFET models for optimal gate sizing/repeater insertion are from Metal Gate/High-K/strained-Si PTM [5].

speed on an optical link to avoid data corruptions during O-to-E conversion.

In Fig. 3.32, we show the circuit schematic of a 2-sink on-chip optical net with drivers and Transimpedance Amplifiers. The 3-pin net is routed with both electrical (Cu) interconnect and a segment of nanophotonic WDM trunk with modulator and photo-detectors. With careful CAD design methodologies, cross-domain opto-electrical interconnect synthesis harbors great potential for both performance(timing) improvement and power reduction.

To evaluate current nano-photonics interconnect status meanwhile achieve forward-looking guidelines for new design methodologies, we model current

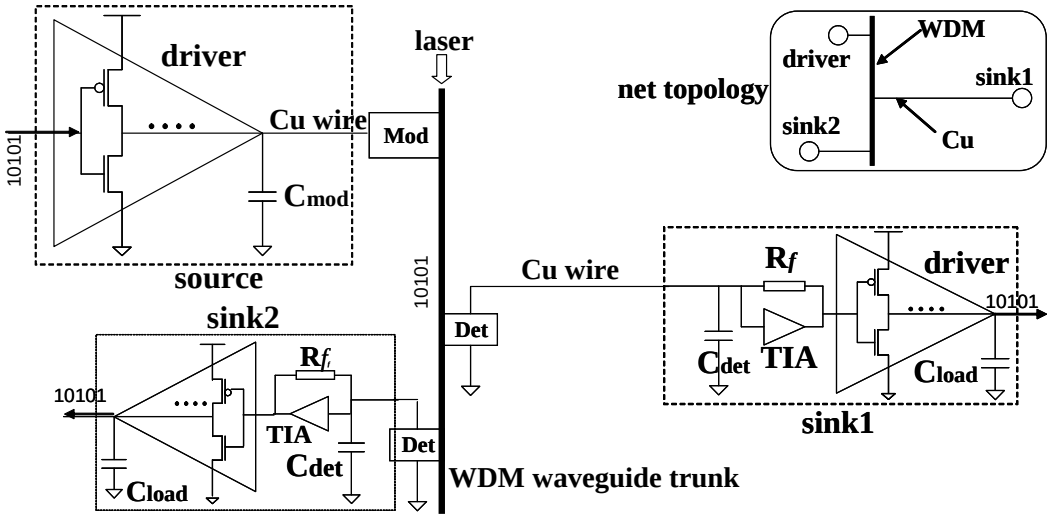


Figure 3.32: Circuit schematic of an on-chip optical net consisted of modulator, WDM waveguide, detectors and driver/amplifier circuits

on-chip scale optical devices and perform comparative analysis with respect to $22nm$ technology global Cu interconnect (metal5/6) with ρ=2.2$\mu\Omega$·cm, R_{sheet}=0.022Ω, C=2pF/cm. CMOS transistor models employed for optimal repeater insertion and gate sizing are from latest $22nm$ node Predictive Technology Model (Metal Gate/High-K/strained-Si) [5]. Further details please refer to Table 3.9.

3.3.2.2 Thermal Reliability Modeling for WDM

Current on-chip WDM techniques mainly fall into the following categories: AWG (array waveguide) structure, ring resonator structure and thin film filter structure. Among these, cavity based ring resonators are most widely employed([65, 78, 98]) due to their compact footprint sizes (allowing for high

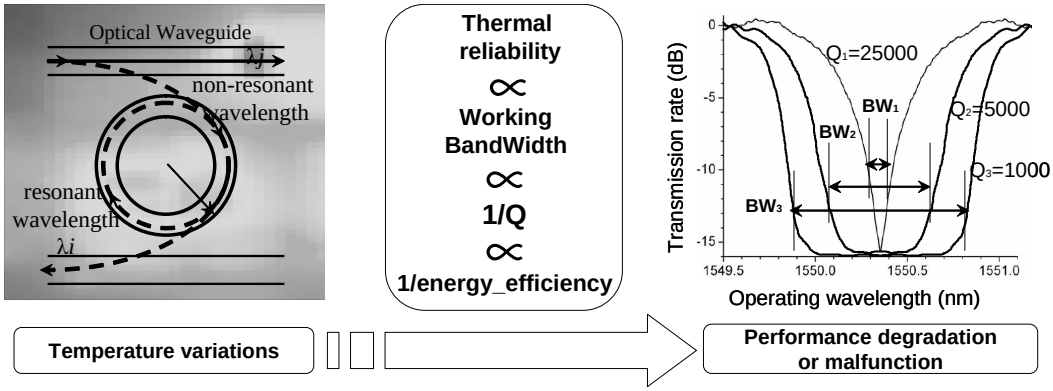


Figure 3.33: Relation between thermal reliability, device bandwidth, quality factor Q and energy efficiency for cavity based optical devices

integration density) and demonstrated high quality factor (Q value). Unfortunately, all nanophotonic devices are prone to thermal variations in their working environment, especially ring resonators.

In particular, on-chip temperature fluctuation causes the central operating frequency (wavelength) of a photonic device to drift. If such a drift results in an off-set that falls outside the range of operating bandwidth (BW), the device will malfunction. Especially for high energy efficiency on-chip UD-WDM devices with ring resonator structure, the quality factor Q [101] (defined as the energy stored in the cavity versus the energy dissipated per unit cycle) is very high and BW is very narrow, rendering the devices highly sensitive to ambience thermal variations. The relationships between thermal reliability, device operating BW, quality factor Q and energy efficiency are defined in Eqn.(3.25)-(3.27) and illustrated in Fig. 3.33.

$$Q = \frac{\lambda_0}{\Delta\lambda_{FWHM}} = \frac{\sqrt{r_1 r_2 a} L \pi n_g}{(1 - r_1 r_2 a)\lambda_0} \tag{3.25}$$

$$n_g(\lambda) = n_e(\lambda) - \lambda \frac{dn_e(\lambda)}{d\lambda} \tag{3.26}$$

$$BW = \triangle f = \frac{f_{resonant}}{Q} \tag{3.27}$$

where r_1, r_2, a, L are ring geometry related parameters, λ_0 is the central working(resonant) wavelength of the ring modulator or detector. n_e is a temperature dependent term, denoting the refractive index of the ring material (e.g., silicon). From the above discussions we can observe that within a relatively small range, one can trade-off Q value for thermal reliability of a certain ring resonator device, without causing aliasing issues in-between of separate channels on a WDM waveguide. However, such a trade-off comes at a power loss penalty that needs to be minimized for power efficient designs.

Based on Eqn.(3.25)-(3.27), we investigate and establish the thermal reliability models for WDM related devices that are mainly based on cavity based components (e.g., ring resonators and ring couplers). The thermal reliability models are obtained through exhaustive temperature dependent refractive index modeling/simulation, working bandwidth characterization, power consumption/dissipation simulation and numerical methods such as Finite-difference Time-domain (FDTD) device simulations on powerful computing platforms using [6]. For further details, please refer to [7].

3.3.2.3 Critical Considerations for On-chip Integration

We investigate the on-chip integration potential of various types of optical links in terms of power, performance(timing) and thermal considerations,

based on the characterized devices and the circuit models.

Considering the calculation of minimal delay on Cu interconnect, we employ the following [129] Eqn. (3.28) to (3.30) to decide the optimal repesater insertion length l and minimal delay τ_e under elmore model, then fine-tune the results with accurate SPICE simulations.

$$l = \sqrt{\frac{2RC'}{R_w C_w}} \tag{3.28}$$

$$\tau_e / l = (2 + \sqrt{2})\sqrt{RC' R_w C_w} \tag{3.29}$$

$$W_{nmos} = \sqrt{\frac{RC_w}{R_w C'}} \tag{3.30}$$

where R, C', R_w, C_w are resistance and capacitance of gate and interconnect respectively. W_{nmos} is NMOS gate width sizing corresponding to the optimal delay. This minimal delay is shown to be about 37ps/mm with finger-structured repeaters inserted every 1.4mm for $22nm$ technology with a 12μm W_{nmos}. More details of wire parameters are shown in Table 3.9.

Considering the delay overhead introduced by E-to-O and O-to-E data conversions, we define *critical length* L_{crit} as the dimension of an on-chip link above which nanophotonics yield shorter signal delay than Cu. Therefore we have Eqn. (3.31):

$$T_{mod} + T_{det} + \tau_o \cdot L \leq \tau_e \cdot L \tag{3.31}$$

where T_{mod} is the E-to-O modulation delay/bit and T_{det} is the O-to-E photo-detection delay/bit; τ_o is signal delay per mm on OWG, τ_e is the delay per mm on Cu interconnect, L is the length of the link. Solving Eqn. (3.31) gives us the range of L, whose lower boundary defines L_{crit} value in mm.

Due to lack of buffering in the optical domain, optical link configuration requires the speed of modulator be upper-bounded by the speed of photo-detection to avoid data corruptions during O-to-E conversions, thus Eqn. (3.32) must hold:

$$T_{det} \leq T_{mod} \tag{3.32}$$

Also for O-to-E conversion, laser power at sink must be equal to or higher than the photo-detection threshold for a logic "1" to be detected successfully, thus Eqn. (3.33) must hold:

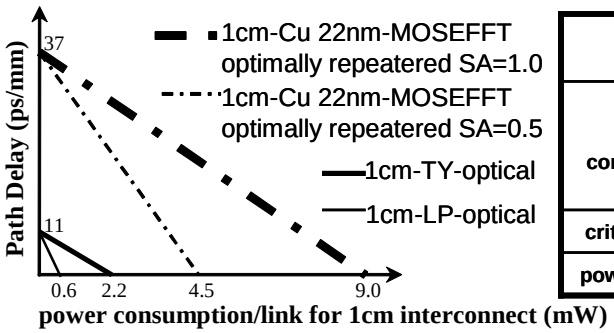
$$P_{O-sink} \geq P_{O-det} \tag{3.33}$$

where P_{O-sink} is the laser power at sink and P_{O-det} is the detector's minimal detection threshold, as listed in Table 3.9.

Considering on-chip temperature variations and the modeling of thermal reliability of the ring resonators, we define $temp\_th$ as the temperature variation threshold value above which a ring resonator based WDM device malfunction. $temp\_th$ corresponds to a scenario in which even trading off the quality factor Q does not compensate the temperature variation, owing to aliased transmission frequencies between different channels of a WDM trunk and related ring modulators/detectors. Therefore, the following Eqn.(3.34)(3.35)(3.36) must hold:

$$Max\_Temp\_Var(trunk_i) \leq temp\_th \tag{3.34}$$

$$Temp\_Var(Ring_i^{mod}) \leq temp\_th \tag{3.35}$$



	Low Power optical link	Typical optical link
link configuration	mod_Ring1	mod_Ring2
	OWG	OWG
	detector1	detector2
critical length	20mm	3.7mm
power/channel	0.6mW	2.2mW

Figure 3.34: Performance comparison - path delay, total opto-electro power per channel - between current on-chip optical links and projected global Cu interconnect (metal5/6) with repeater insertion for a local clock f_{clk}=5GHz with τ_{slew}<10ps. Optical link configuration parameters are shown in the tabl. Only dynamic power on interconnect is considered for electrical links

$$Temp\_Var(Ring_j^{det}) \leq temp\_th \tag{3.36}$$

In Eqn.(3.34), all the nodes along the path of a WDM trunk must satisfy the $temp\_th$ condition; while in Eqn.(3.35)(3.36), both modulation and detection ring resonators on link node$i$→node$j$ must also meet $temp\_th$. Therefore with Eqn.(3.34)(3.35)(3.36), the whole optical link's thermal reliability constraint is properly set.

Utilizing afore-mentioned characterizations, we briefly illustrate the performances (delay and power consumption) of $1cm$ nanophotonincs and Cu on-chip links in Fig. 3.34, where smaller delay-power products (triangle areas under the lines) represent better overall performance. Switching Activities of the global Cu links are set to 50% and 100% respectively with a local f_{clk} of 5GHz; the optical links are configured as LP (low power) and TY (typical)

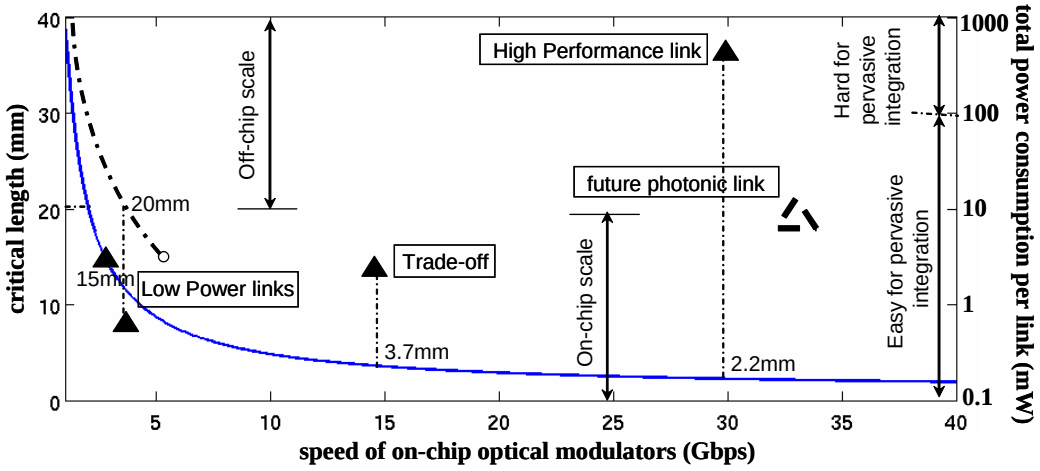


Figure 3.35: A chart analysis for various configured optical links with currently demonstrated nanophotonic devices

with details listed in the table of Fig. 3.34. From the calculated *critical length* of the two optical links listed, we can see that LP is an ideal candidate link configuration for low power photonic Network-on-Chips, while TY allows for more physical design flexibilities/capabilities for on-chip integration synthesis in e.g, the routing stage.

We calculate the critical lengths and corresponding power consumptions (optical and electrical) of differently configured *optical links* versus a continuous axis of optical modulation speed in Giga bit per second unit in Fig. 3.35. The curves represent critical length versus speed of modulation relation, where the black dotted curve uses *detector1* (5Gbps) at sink (so the curve ends at 5Gbps modulation) and the blue curve uses *detector2* (40Gbps) at sink. The black triangles are differently configured optical links consisted of modulators, OWG and photo-detectors of various types. The height of each

triangle represents the total optical and electrical power required to drive a single $1cm$ long on-chip photonic link. Therefore, Fig. 3.35 gives us the critical length (left-hand side y-axis) versus total power consumption (right-hand side y-axis) relation for each configured type of on-chip optical links based on current technology.

From Fig. 3.35, we observe that: (1) current technology allows feasible low power optical links (on the scale of $15mm$-$20mm$) to replace projected $22nm$ technology global Cu interconnect with positive gain in both signal speed and total link power consumption, making a very promising global interconnect alternative; (2) the *critical length* L_{crit} for currently feasible on-chip optical interconnect physical design is about $3.7mm$; (3) considering nanophotonics advancement within the near future, on-chip photonics links will mainly dominate in the range of $2mm$ to $20mm$ interconnection with positive signal speed gain than the projected global Cu interconnect. In the following subsections, we will explore the design space from CAD physical design perspective using the typical "trade-off" configurations of optical links.

3.3.3 Overall CAD Flow

We present a CAD framework for low power thermal-aware on-chip WDM integration, using the models from Subsection 3.3.2.

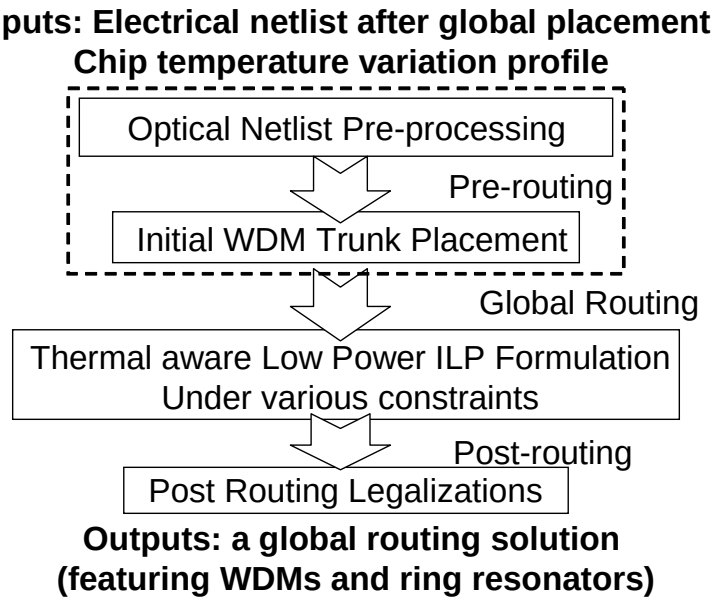


Figure 3.36: An overview of our proposed CAD flow

3.3.3.1 Overview

In Fig. 3.36, we illustrate a top level flow diagram of our proposed method, starting from a given input netlist and on-chip temperature variation profile. Such a CAD flow is consisted of 3 major stages: a **Pre-routing** stage that prepares the optical netlist and WDM trunk placement; a **Global Routing** stage that serves as the core formulation of the WDM channel assignment problem based on various physical design constraints; and a **Post-routing** stage that further examines legalization related issues in both the optical and electrical domains. We describe each function block of Fig. 3.36 in detail as follows.

3.3.3.2 Netlist Pre-processing

Netlist pre-processing step prepares the optical netlist with an initial consideration of the *timing condition* which guarantees that the circuit timing does not degrade after employing nanophotonics (since each data conversion takes significant time). This step is mainly proposed to derive optical netlist test cases from existing electrical benchmarks such as ISPD global routing netlists. This step is very critical since it selects proper pins (nets or partial nets) from the electrically placed netlist to synthesize in the **Global Routing** stage. The selection is designed such that the minimal manhattan distance of all driver-sink pairs mapped onto the optical domain is lower bounded by the *critical length* L_{crit}. This step serves to yield *non-negative timing gain* in the optical domain than in the electrical domain. This aligns well with *critical length* definition and discussions in Subsection 3.3.2. The main technique involved is described as follows,

Pin Clustering: To cluster the electrically placed input netlist based on manhattan distance using hierarchical clustering method. In this case, we first construct the *dendrogram* (illustrated in Fig. 3.37) and then pick out the clusters satisfying the L_{crit} dimension with a *depth first search* on the *dendrogram*. The result of this procedure is a set of clusters whose respective geometric medians are mapped to the optical domain as pseudo-pins. These pseudo-pins form the *Optical Netlist*, while the rest of pins within each cluster remain on the electrical domain and are electrically interconnected to their geometric median. Therefore, only 1 O-to-E or E-to-O conversion is needed

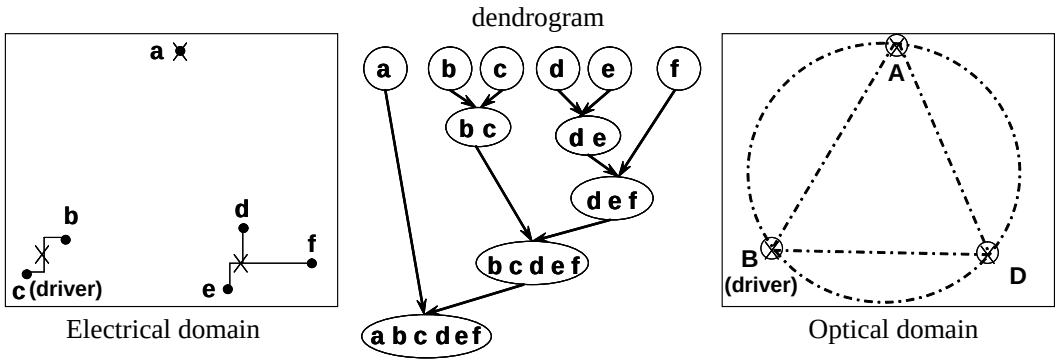


Figure 3.37: A brief illustration of netlist pre-processing

per cluster. This procedure is briefly illustrated in Fig. 3.37, where a-f are pins of certain net in the electrical netlist and ABD are pseudo pins (a partial net) mapped onto the optical plane to represent clusters with edges larger than L_{crit} in the *dendrogram*. B is the driver pin in optical domain since driver pin c lies in the bc cluster in electrical domain.

3.3.3.3 Initial WDM Trunk Placement

Initial WDM trunk placement depend on the median of geometry distributions of optical nets in the *Optical Netlist* and is carried out in a partitioned manner across the whole chip area according to Eq. (3.37), until the total number of WDM channels is sufficient to hold the total number of optical nets/links in the netlist.

$$Place_{trunk^k} = med\{med[net\_i]\}^{i \in Partition^k} \tag{3.37}$$

Our partitioning based initial placement is carried out in 4 steps as illustrated in Fig. 3.38:

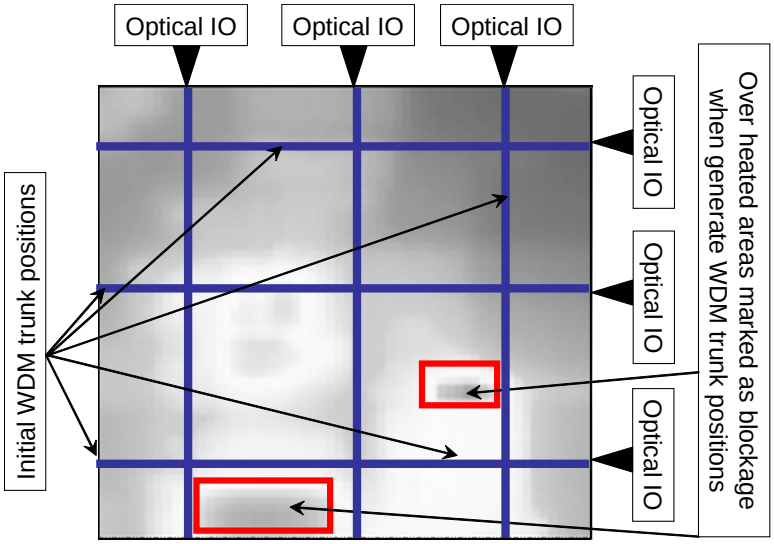


Figure 3.38: Illustration for the WDM OWG initial placement

- Continues for both horizontal and vertical directions
- Avoids over-heated regions marked as thermal blockages
- Partition ends when the number of WDM channels are sufficient for the total number of links in the optical netlist
- Extra WDM trunks may need to be added in the **Post-routing** stage, depending on the feasibility of the optimal solution

3.3.3.4 Thermal-aware Low Power Routing

Thermal-aware low-power driven synthesis takes place in the **Global Routing** stage as shown in Fig. 3.36. Our proposed routing techniques are based on the following rule-of-thumbs:

- Multiple optical waveguides (OWG) are placed on a chip with optical IO ports and off-chip laser sources
- Each OWG has N number of total channels, with each channel assigned a unique wavelength λ as its working band
- O-E and E-O data conversions (modulation and detection) happen along the trunks of OWGs
- Couplings and crossings of OWGs result in power loss thus more laser power required to compensate the loss
- Paths/links on the same net are encouraged to be routed on the same channel of certain OWG
- Different paths/links from different nets must be routed on different OWGs or different channels of a certain OWG to avoid shorted paths
- WDM trunks that are not eventually assigned any nets will be turned off (no laser source input from off-chip)
- WDM channels that are not eventually assigned any data links will be turned off (no laser source input from off-chip)
- Legalizations are mainly left to the **Detailed Routing** stages

Here we define the *timing condition* as the condition that guarantees smaller signaling delay on the opto-electrical link than on Cu interconnect.

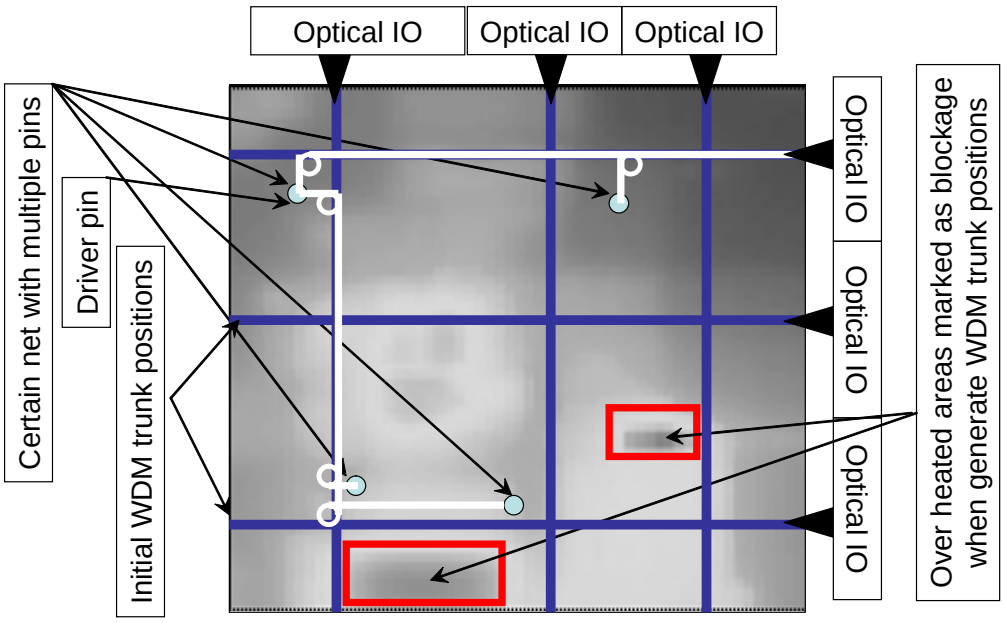


Figure 3.39: Our WDM based global routing scenario

This is a critical consideration since each additional O-E/E-O data conversion brings significant delay. The *thermal condition* is defined to make sure the local temperature variation does not fall out of the working range of the ring modulators. In case of a violated *thermal condition*: (1) Q value will be adjusted to trade-off power efficiency for thermal reliability; (2) if (1) can not be done without causing aliases between separate WDM channels, that particular region is set as a thermal blockage. For the core routing problem of the **Global Routing** stage we propose 2 approaches: *CAT* and *GLOW*.

In Fig. 3.39, we illustrate the routing problem after the **Pre-routing** stage, with laser sources from off-chip whose driving power to each WDM waveguide trunk differs according to the total number of channels assigned/utilized

after the routing stage. To constrain the solution space for the global routing stage, we assume to take the shortest distance route when a pin is to connect to certain WDM trunk, i.e, data conversion (modulation/detection) only happens on WDM trunks. Based on Fig. 3.39, we will briefly discuss both approaches as follows, details will be presented in Subsection 3.3.4 and 3.3.5.

CAT: A greedy heuristic approach for WDM Channel Assignment under Thermal considerations. The basic motivation for *CAT* is to assign optical nets/links to WDM trunks in a sequential manner, meanwhile to combine timing and thermal-awareness constraints locally for each WDM trunk. In particular, *CAT* picks all the local nets/link satisfying the timing condition and assign the least power consuming links to fill the available channels to certain WDM waveguide, then move onto the next waveguide. If at the end of the process, there are still unassigned nets, then the *Initial WDM Placement* stage will be appended with extra WDM resources to route the remainder nets.

CAT's advantage mainly include run time and the simplicity of the implementation. However, key power related factors are neglected such as WDM trunk crossings and the co-relation between thermal reliability and Q value related resonant power loss. Also, there is no global guarantee that the final solution reached is minimal power. We present an alternative approach as an improvement over *CAT* to address these disadvantages:

GLOW: An ILP based global routing approach for low power driven thermal-reliable WDM synthesis. *GLOW* is a low power driven global router with various physical design constraints. With careful selection of IV (integer

variables) and BV (binary variables), we not only formulate the key power related terms, but also the cross-related variables and constraints that are otherwise very hard to capture. We will discuss its mathematical formulations in Subsection 3.3.5.

3.3.3.5 Post Routing Legalization

This stage is mainly to resolve the cases when multiple rings are contending the same geometry location, which causes design rule violations in the optical and electrical domains. For this work, we use simple perturbation based re-routing/adjustment techniques for legalizations in the cross domain of optoelectronics. Further considerations of post routing stages legalizations will be left to detailed routing stages.

3.3.4 CAT Routing Algorithm

CAT is designed and implemented as a greedy heuristic approach for thermal-aware WDM channel assignment under timing constraints. It is performed in 3 major steps: first, *Initial WDM Trunk Placement*; second, *Timing and Thermal Condition Calculation*; third, *Greedy Channel Assignment*. We describe each step in detail as follows,

Initial WDM Trunk Placement: *CAT* uses the same trunk initial placement result as in *GLOW*.

Timing and Thermal Condition Calculation: In this step, all the WDM trunks are traversed in certain order sequentially. For each trunk,

timing/thermal conditions for all optical links are calculated and updated using models from Subsection 3.3.2.

Greedy Channel Assignment: For the channel assignment, we use a greedy heuristic method that executes in 3 phases: **Phase1**: Form set $S(link_i)$ for WDM $trunk_i$ with the optical links that guarantee smaller signaling delay than in the electrical domain. $S(link_i)$ is a set of link candidates to be assigned to WDM $trunk_i$. **Phase2**: Sort the links in $S(link_i)$ with *Thermal Condition* metric in ascending order. **Phase3**: Assign links from $S(link_i)$ to $trunk_i$ in ascending order, until the total number of optical nets assigned reaches $Cmax$. For more details of *CAT*, please refer to Algorithm 16.

3.3.5 GLOW Routing Algorithm

3.3.5.1 ILP Formulation

To formulate the optical routing problem, we introduce parameters and binary/integer variables in Table 3.10. We elaborate a few of them here:

• n, m: total number of WDM trunks in the row and column directions after initial placement, respectively.

• W_i: binary variables denoting the assignment status of WDM trunk i. If W_i is 0, trunk i is not unassigned any optical nets in the final routing solution, therefore will not be turned on (no input laser power from its optical IO port); if W_i is 1, trunk i is assigned certain nets, but may still has available channels.

Algorithm 16 *CAT*: Channel Assignment for Thermal-reliability

Require: (1) Initial WDM trunk placement
Require: (2) Temperature variation profile
Require: (3) Optical netlist
  Generate $L(link)$ as a set of all unassigned links/nets
  for each WDM $trunk_i$ **do**
    for each $link_j$ in the optical netlist **do**
      Calculate timing constraint on ($link_j$,$trunk_i$)
      Calculate thermal variation constraint on ($link_j$,$trunk_i$)
    end for
    Form set $S(link_i)$ by links that satisfy *Timing Condition*
    Sort $S(link_i)$ based on ascending *Thermal Variation*
    Select $link_k \in S(link_i)$ in ascending index order
    while $link_k$ unassigned AND $trunk_i$ has available channels **do**
      Assign $link_k$ to WDM $trunk_i$
      Update set $L(link)$
    end while
  end for
  Form set $U(link)$ with still unassigned links in $L(link)$
  if set $U(link) = \emptyset$ **then**
    Prompt: all links assigned
  else
    Add new WDM trunks to *Initial Placement*
    Modify *Optical Netlist*
    Call $CAT(\cdot)$ recursively
  end if
  return WDM channel assignment AND optical/laser power

Table 3.10: Variables/parameters in ILP formulation

Name	**Description**
P_{total}	total laser power consumed
P_{loss}	total on-chip laser power loss
$P_{dynamic}$	total on-chip laser power for optical signaling
P_0	base power consumption for a WDM trunk
P_{cross}	total power loss due to trunk crossings
P_{trunk_thm}	total power loss due to trunk thermal effects
P_{ring_thm}	total power loss due to ring thermal effects
P_{path}	total power loss due to photon propagation
$P_{\lambda i}$	laser power on channel λi for optical signaling
P_{thm}^{ij}	laser power loss when trunk i, trunk j cross
$P_{trunk\_thm}^{i}$	thermal related power loss on trunk i
$P_{ring}^{link_i}$	laser power loss on the rings of link i
W_i	BV: allocation status of trunk i
W_{ij}	BV: crossing status of trunk i and trunk j
$S_{link_i}^{trunk_j}$	BV: assignment status of link$i$ onto trunk j
$Sum_{net_i}^{trunk_j}$	IV: # of links in net i assigned to trunk j
$\lambda_{net_i}^{trunk_j}$	BV: assignment status of net$i$ onto trunk j
$T_{var}^{link_i}$	temperature variation on the rings of link i
C_{max}	channel capacity of each WDM trunk
PIN_{max}	max pin # in certain net of the optical netlist
$temp\_th$	temperature variation tolerance threshold
τ_e	delay per unit length on Cu interconnect
τ_o	delay per unit length on optical links
τ_{conv}	delay overhead by data conversions
WL_e^i	Cu wire length on link i
WL_o^i	optical wire length on link i
$HPWL^{link_i}$	half parameter wire length of link i

• W_{ij}: binary variables numerically equal to the product of W_i and W_j, where $i \in [0, n-1]$, $j \in [n, n+m-1]$. If W_{ij} is 0, trunk i and trunk j are not physically crossed, vise versa.

• $S_{link_k}^{trunk_i}$: binary variables, with 0 meaning link k is assigned onto WDM trunk i.

• $Sum_{net_i}^{trunk_j}$: integer variables, representing the total number of optical nets assigned onto trunk j in the final solution.

• $\lambda_{net_i}^{trunk_j}$: binary variables, with 0 meaning net i is assigned onto WDM trunk j in the final routing solution; vise versa.

• $Cmax$: channel capacity of each WDM trunk. It is total available channel number that serves at an upper bound limit condition when assigning optical nets.

• $PINmax$: max pin number of certain net in the optical netlist. For our proposed formulation, $PINmax$ can take any number.

Please see Table 3.10 for a complete list of parameters/variables. With these parameters, we propose the following objective function for $GLOW$'s thermal-aware low power routing with on-chip WDM:

$$Minimize\{P_{total}\} \ \ w.r.t \ \ W_i,\ W_{ij},\ S_{link_i}^{trunk_j},\ \lambda_{net_i}^{trunk_j} \tag{3.38}$$

such that:

$$P_{total} = P_{loss} + P_{dynamic} \tag{3.39}$$

$$P_{loss} = P_{cross} + P_{trunk\_thm} + P_{ring\_thm} + P_{path} \tag{3.40}$$

$$P_{cross} = \sum_{i \in [0,n-1]}^{j \in [n,n+m-1]} W_{ij} * P_{thm}^{ij} \tag{3.41}$$

$$P_{trunk\_thm} = \sum_{i}^{i \in all\ trunks} W_i * P_{trunk\_thm}^{i} \tag{3.42}$$

$$P_{ring\_thm} = \sum_{i}^{i \in all\ trunks} \sum_{j}^{j \in all\ links} S_{link_j}^{trunk_i} * P_{ring}^{link_j} \tag{3.43}$$

$$P_{dynamic} = \sum_{i}^{i \in all\ trunks} \sum_{j}^{j \in all\ nets} \lambda_{net_j}^{trunk_i} P_{\lambda i} + \sum_{i} W_i P_0 \tag{3.44}$$

Eq. (3.38) above gives the objective function of *GLOW* as the total power P_{total} required to drive the circuit. As shown in Eq. (3.39), P_{total} is divided into 2 parts: the total optical power loss on chip P_{loss}, which is the amount of power the drivers need to compensate for the guarantee of *detection conditions* on photo-detectors; and $P_{dynamic}$, the signal switching power on WDM channel carriers.

P_{loss} is divided into 4 terms: waveguide crossing power, thermal related WDM trunk power, thermal related ring resonator power and the power to compensate propagation loss of on-chip waveguide.

$P_{dynamic}$ consists of 2 terms: P_0 is the base power consumption for each WDM trunk, it is a constant power cost when turning on a N-channel WMD trunk; the 2nd term is the switching power on all WDM channels, which is linearly proportional to the number of channels utilized. Apparently, WDM trunk multiplexing/sharing rate is to be maximized in order to avoid unnecessary P_0's.

All power related terms are modeled according to our previous discussions in Subsection 3.3.2 and Subsection 3.3.3, please also see Table 3.10 for further explanation of each term.

3.3.5.2 Physical Design Constraints

Following the discussions in Subsection 3.3.2, we present the detailed mathematical expressions of the 6 types of constraints employed in the formulation of *GLOW*:

• Timing constraint: for each optical link, the routing solution must not result in longer signal delay than HPWL estimated delay in the electrical domain:

$$S_{link_i}^{trunk_j}[\tau_e * WL_e^i + \tau_o * WL_o^i + \tau_{conv}] \leq \tau_e * HPWL^{link_i} \tag{3.45}$$

• Selection constraint: to make sure each link i is only assigned to one WDM trunk. For each link i, we have the following:

$$\sum_{j}^{j \in all\ trunks} S_{link_i}^{trunk_j} = 1 \tag{3.46}$$

• Channel capacity constraint: to make sure each WDM trunk does not exceed its capacity limit.

For each WDM trunk j:

$$\sum_{i}^{i \in all\ nets} \lambda_{net_i}^{trunk_j} \leq Cmax \tag{3.47}$$

After the global routing stage, the term $\sum_{i}^{i \in all\ nets} \lambda_{net_i}^{trunk_j}$ equals to the total number of WDM channels assigned onto WDM trunk j. According to our rule-of-thumbs previously described in Subsection 3.3.3.4, the unassigned channels (corresponding to zero valued λ_{net}^{trunk}) will be turned off from off-chip laser sources to reduce power consumptions.

• Detection constraint: the final optical power at each sink on each link must be large enough to be detected by the photo-detectors.

• Thermal constraint: for each link (pair of pins from source to certain sink), local temperature variation must not result in performance degradation or malfunction. Therefore for each link i and trunk j:

$$S_{link_i}^{trunk_j} * T_{var}^{link_i} \leq temp\_th \tag{3.48}$$

• Binary/Integer variable constraints: since W_{ij} and $\lambda_{net_i}^{trunk_j}$ are introduced to eliminate non-linear terms under our proposed Integer Linear Programming formulation, the following extra constraints must also be enforced:

$$2W_{ij} \leq W_i + W_j \leq 1 + W_{ij} \tag{3.49}$$

where $i \in [0, n-1]$, $j \in [n, n+m-1]$

$$\frac{(2\sum_{k}^{k \in net_i} S_{link_k}^{trunk_j} - 1)}{2PINmax} \leq \lambda_{net_i}^{trunk_j} \leq 2 \sum_{k}^{k \in net_i} S_{link_k}^{trunk_j} \tag{3.50}$$

$$\frac{(2\sum_{i=1}^{all\ nets} \lambda_{net_i}^{trunk_j} - 1)}{2Cmax} \leq W_j \leq 2 \sum_{i=1}^{all\ \ nets} \lambda_{net_i}^{trunk_j} \tag{3.51}$$

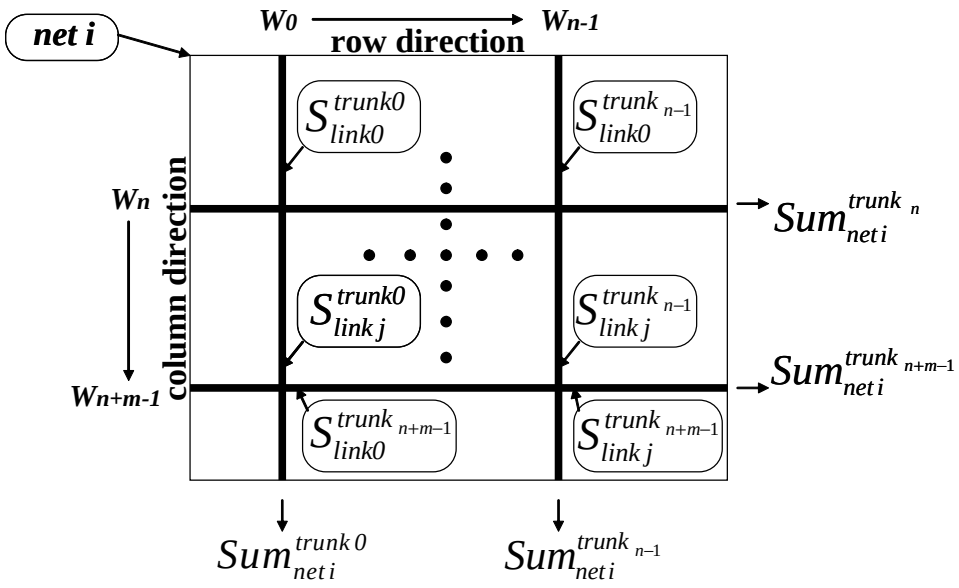


Figure 3.40: Relation between $S^{trunk_j}_{link_k}$ and $Sum^{trunk_j}_{net_i}$

Here Equation(3.50) and (3.51) are enforced for two-fold reasons: (1) we are able to calculate the number of optical nets assigned to certain WDM trunk via optical link related variables; (2) to introduce non-linear relation between $\lambda^{trunk_j}_{net_i}$ and $S^{trunk_j}_{link_k}$ under ILP formulation. For this part an intermediate term $Sum^{trunk_j}_{net_i}$ is introduced by Equation(3.52) as follows,

$$Sum^{trunk_j}_{net_i} = \sum_{k}^{k \in net_i} S^{trunk_j}_{link_k} \tag{3.52}$$

Equation(3.50)(3.51)(3.52) together make sure that if $Sum^{trunk_j}_{net_i} = 0$, then $\lambda^{trunk_j}_{net_i} = 0$; if $Sum^{trunk_j}_{net_i} > 0$, then $\lambda^{trunk_j}_{net_i} = 1$. Further illustration of the relation between $S^{trunk_j}_{link_k}$ and $Sum^{trunk_j}_{net_i}$ is shown in Fig. 3.40.

Please refer to Algorithm 17 for more details on *GLOW*.

Algorithm 17 *GLOW*: Global Routing for Low Power WDM

Require: (1) Initial WDM trunk placement
Require: (2) Temperature variation profile
Require: (3) Optical netlist
  for each WDM $trunk_i$ **do**
    for each optical $link_j$ **do**
      Calculate P_{cross}, $P_{WDM\_thermal}$, $P_{ring\_thermal}$
      Calculate $P_{dynamic}$
      Update *Timing Constraint*, *Thermal Constraint*
    end for
  end for
  Invoke ILP solver
  return WDM channel assignment AND optical/laser power

3.3.6 Experimental Results

CAT and *GLOW* are implemented and assessed by various testing benchmarks derived from ISPD global contest netlists. We describe the benchmark preparation and discuss/compare the simulation results as follows,

Benchmarks and Simulation Setups: In Table 3.11 we list 6 benchmarks: IBM01-06, with net number ranging from 35 upto 996. These test cases are derived from IPSD global routing contest benchmarks by: (1) up-scaling the chip dimension into centimeter scale; (2) employing our proposed *Optical Netlist Pre-processing* techniques to generate optical netlists. Considering the limited integration volume of current on-chip WDM nanophotonics, the sizes of these testing netlists are suitably representative.

For the hierarchical clustering procedure, L_{crit} is set to 3.7mm for centimeter-scale chips. We assume all the inserted ring resonators are legal-

ized and initially thermally tuned. The on-chip thermal variation profiles are randomly generated based on measured data of real processor chips. The tolerance threshold $temp\_th$ of the maximal range of temperature variation is set to between 15 to 20 degrees, as hard constraints in our problem formulation. For the WDM trunk initial placement, we use 32-channel WDM trunks to start with, then run the proposed global routing algorithms on 3.0GHz Linux workstations with 8GB memories.

Result and Analysis: In Table 3.11, we show simulation results of *CAT* and *GLOW*, with total power consumption normalized to the power value that *GLOW* gives on IBM01. Compared with *CAT*, *GLOW* demonstrates significant 22%-49% of total power reductions on IBM01-06, respectively.

Reasons of such improvement are mainly two-fold: first, *CAT* only searches for local optimal solutions and assign optical nets/links to WDM trunks in a sequential/local manner, while *GLOW* aims at a global optimal solution with mathematical programming techniques; second, *CAT* is not aware of the waveguide crossing power, nor does it consider the thermal related ring resonator power-reliability trade-off in a global manner; while on the other hand, the ILP formulation of *GLOW* makes it possible to model all the key power contributors.

Also in Table 3.11 we show the WDM channel/trunk allocation status of *CAT* and *GLOW* on IBM01-06, together with a comparison chart of the average number of assigned channels per trunk, as depicted in Fig. 3.41. We see that compared with *GLOW*, *CAT* assigns fewer number of WDM trunks,

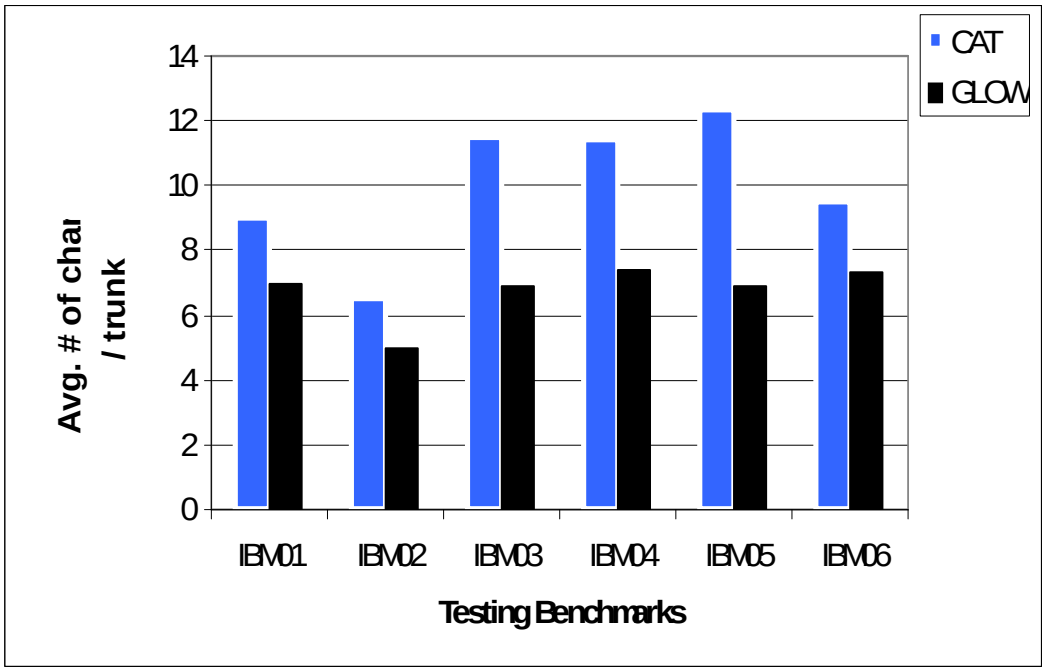


Figure 3.41: A comparison chart of average number of assigned WDM channels per WDM trunk, between *CAT* and *GLOW*

resulting in a slightly higher number of average WDM channels per trunk and shorter total length of on-chip WDM waveguide. *GLOW*, however, works by assigning WDM trunks/channels across the chip aiming at the global solution of power consumption minimization under given thermal reliability requirements. This helps *GLOW* to bring down the total power at the cost of some extra OWG wirelength. This is acceptable since the fabrication cost of straight OWGs are relatively low meanwhile the silicon layer provides rich resources for monolithic integration of the required nanophotonics components.

Table 3.11: Simulation result comparisons between our proposed *CAT* and *GLOW*

Algorithm	**CAT**						**GLOW**					
Test Case	IBM1	IBM2	IBM3	IBM4	IBM5	IBM6	IBM1	IBM2	IBM3	IBM4	IBM5	IBM6
Net #	35	70	137	240	437	996	35	70	137	240	437	996
Pin #	95	187	391	658	1357	2698	95	187	391	658	1357	2698
Sink #	60	117	254	418	920	1702	60	117	254	418	920	1702
Trunk #[a]	4	11	12	25	46	138	5	16	22	40	87	193
Channel #[b]	36	72	138	286	570	1314	35	79	152	295	602	1408
Avg. channel versus trunk	9.0	6.55	11.5	11.44	12.39	9.52	7.0	4.94	6.9	7.38	6.92	7.29
Total trunk length cm	4.8	13.2	14.4	30	55.2	165.6	6.0	19.2	26.4	48.0	104.4	231.6
Total power[c]	1.45	4.68	6.81	13.8	27.26	65.52	1.00	2.48	5.27	7.25	16.63	32.86
Avg. power reduction %	-	-	-	-	-	-	31.0%	47.0%	22.6%	47.5%	39.0%	49.8%

[a] Each WDM trunk has a maximum of 32 available channels in the initial placement stage. Unassigned trunks will be turned off in the global routing stage.

[b] Unassigned WDM channels will be turned off (no laser input from off-chip) in the global routing stage.

[c] Total power consumption is normalized to the power consumed on IBM01 by *GLOW*.

In a few cases when there are no feasible solutions exist, the ILP formulation will not return valid WDM channel/trunk allocation strategy and the WDM trunk initial placement must be adjusted (by adding more trunks). Such adjustments are carried out in a progressive and heuristic manner until feasible integer solutions are found. Compared with *CAT*, *GLOW* gives equal or acceptable longer run-time on all 6 benchmarks. Considering the growth of the problem complexity and relatively limited on-chip photonics integration volume in the near future, the CPU time overhead by *GLOW* is well acceptable.

3.3.7 Summary

In this section, we present *GLOW* to examine the integration potential and explore the design space of low power, thermal reliable on-chip nanophotonics interconnect featuring Wavelength Division Multiplexing (WDM) mechanism. As an Integer Linear Programming based global routing strategy, *GLOW* aims at low power on-chip WDM integration under the considerations of thermal reliability modeling and various physical design constraints such as power(thermal), timing and signal quality. *GLOW* is simulated and evaluated on various testing cases derived from ISPD global routing contest benchmarks. Compared with an alternative approach: *CAT* - a thermal-aware heuristic channel assignment method, *GLOW* demonstrates 22%-49% of total power reduction, revealing great design automation potential towards future on-chip giga-scale nano-photonics WDM integration. Since this is the first

study, to our best knowledge, to link CAD and Physical Design together with nanophotonics WDM modeling and simulations, we believe a lot of future researches can be done to co-optimize the CAD and nanophotonics technologies.

Chapter 4

Conclusion

In this dissertation, we studied CAD optimization techniques for high-yield nanometer IC design under advanced nanolithography technologies and state-of-the-art Resolution Enhancement Techniques (RETs) in Chapter 2. Based on these discussions, we explored the applications of nanolithography manufacturing technologies in the emerging field of on-chip Silicon compatible nanophotonics in Chapter 3. We explored the physical design space of on-chip nanophotonics interconnect for next generation low-power, high-performance and thermal-reliable opto-electrical ICs projected at advanced lithography nodes (e.g., sub-22nm node).

4.1 VLSI CAD for Nanolithography

As crucial assistance for simulation and optimization for advanced nanolithography technology, we investigate fast, accurate and reliable lithographic hotspot detectors in Chapter 2. In Section 2.2 we presented a *critical feature* extraction/classification based hotspot detection flow utilizing modern machine learning (artificial neural network) technique. With *critical feature* representation, learning/detection noise for the training procedure was effectively

reduced without run-time overhead when compared with [92]. Experimental results demonstrated small detection false-alarms (10% of actual hotspots) and 90% detection accuracy on average, with best achieved accuracy 100%.

To alleviate the run-time cost of lithographic simulations and further improve the accuracies, in Section 2.3 we proposed an ultra-fast and high fidelity hotspot detection flow providing full layout, feature-centric assessment as improvement over sliding window or raster scanning techniques. Under the real manufacturing conditions, we incorporated a novel set of hotspot signature measurement, a hierarchically refined classification methodology and powerful machine learning kernel implementations into an integrative flow. We implemented our algorithm with an industry-strength engine [1] under real manufacturing conditions, and showed that it significantly outperforms previous state-of-the-art algorithms in hotspot detection false alarm rate (2.4X to 2300X reduction) and simulation run-time (5X to 237X reduction), meanwhile archiving similar or slightly better hotspot detection accuracies. The demonstrated high performance makes our approach very suitable for identifying lithographic hotspots and guiding lithography-friendly physical design.

In Section 2.4, we proposed a generic and unified meta-classification framework to combine the strengths of various disparate hotspot detection techniques. Different machine learning techniques and pattern matching methods were developed and experimented under the proposed framework, which proved to achieve very impressive capability to trade-off between hotspot detection accuracies and false-alarm suppressions.

In Section 2.5, we explored the application of hotspot detection engines in the early design stages, namely the detailed routing stage. We developed a fast and generic formulation for a manufacurablity-friendly detailed router. Our proposed formulation out-performs previous state-of-the-art lithography-aware routers, meanwhile maintains fast CPU run-time.

With these explorations, we have demonstrated the critical and effective role of CAD techniques in addressing the many DFM challenges under the nanolithography process technologies. Over the years, we expect to see more innovations and emerging techniques along this direction as CAD methodologies are employed in achieving optimized vertical design integrations.

4.2 VLSI CAD for Nanophotonics

To leverage the nanolithography manufacturing technologies, we study the design and optimization of a new on-chip interconnect technology using Silicon compatible nanophotonics devices in Chapter 3.

In Section 3.1, we proposed OIL (Optical Interconnect Library): a fully characterized collection of silicon nano-photonic devices for system level interconnect planning/analysis and low power high performance design/synthesis explorations towards a new *holistic photonic Networks-on-Chip* paradigm.

With OIL, we presented in Section 3.2 the first optical routing framework, *O-Router* for low power on-chip integration of silicon nano-photonics with consideration of various detection constraints. Based on ILP formulation

with several variable reduction techniques for routing speed-ups, *O-Router* utilizes OIL with key parameters projected for future technologies based on optical interconnect roadmap. Experimental results show promising improvements compared with traditional Minimum Spanning Tree routing algorithm.

In Section 3.3, we further examined the thermal-reliability issues of the on-chip nanophotonics devices and propose a CAD flow to achieve low-power thermal-reliable design integration through utilizing on-chip optical Wavelength Division Multiplexing devices.

With the above explorations and discussions, we have demonstrated the unique role of CAD optimization techniques in the process of the design and optimization of future low-power thermal-reliable nanophotonic ICs under advanced nanolithography manufacturing technology. With the many more challenges to address in this emerging field, we expect to see a lot of future works along this direction as new nanophotonics devices are introduced for the ultimate global optical and electrical interconnect co-synthesis and planning.

Bibliography

[1] CALIBRE Mentor Graphics Corp.

[2] http://www.nangate.com, NANGATE 45nm Cell Library.

[3] http://www.tela-inc.com,1D Restrict Design Rule.

[4] MATLAB MathWork Corp.

[5] Predictive Technology Model, http://ptm.asu.edu.

[6] RSoft Photonics CAD Suite version 5.1.7, by RSOFT Inc.

[7] http://www.cerc.utexas.edu/~ding/oil.

[8] International Technology Roadmap for Semiconductors. 2010.

[9] Mandeep Bamal, Scott List, Michele Stucchi, Anne S. Verhulst, Marleen Van Hove, Rudi Cartuyvels, Gerald Beyer, and Karen Maex. Performance Comparison of Interconnect Technology and Architecture Options for Deep Submicron Technology Nodes. In *International Interconnect Technology Conference*, 2006.

[10] C. Bencher. SADP: The Best Option. In *Nanochip Technology Journal*, 2007.

[11] Luca Benini and Giovanni De Micheli. Networks on Chip: A New Paradigm for System on Chip Design. In *Proc. Design, Automation and Test in Eurpoe*, 2002.

[12] Keren Bergman. Silicon Photonic On-Chip Optical Interconnection Networks. In *Frontiers in Nanophotonics and Plasmonics*, Guaruja, Brazil, Nov 2007.

[13] Aleksandr Biberman, Benjamin G. Lee, Keren Bergman, Amy C. Turner-Foster, Michal Lipson, Mark A. Foster, and Alexander L. Gaeta. First Demonstration of On-Chip Wavelength Multicasting. In *Optical Fiber Communication Conference*, March 2009.

[14] Wim Bogaerts, Pieter Dumon, Dries Van Thourhout, Dirk Taillaert, Patrick Jaenen, Johan Wouters, Stephan Beckx, Vincent Wiaux, and Roel G. Baets. Compact Wavelength-Selective Functions in Silicon-on-Insulator Photonic Wires. In *IEEE Journal of Selected Topics in Quantum Electronics*, Dec. 2006.

[15] Mark Bohr. Silicon Technology Leadership and the New Scaling Paradigm. In *Intel Developer Forum*, April 2007.

[16] Y. Borodovsky. Lithography 2009 Overview of Opportunities. In *Semicon West*, 2009.

[17] A. C. Bovik, editor. *Handbook of Image and Video Processing*. Academic Press, 2000.

[18] H. Breu, J. Gil, D. Kirkpatrick, et al. Linear Time Euclidean Distance Transform Algorithms. In *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 1995.

[19] Johnnie Chan, Aleksandr Biberman, Benjamin G. Lee, and Keren Bergman. Insertion Loss Analysis in a Photonic Interconnect Network for On-Chip and Off-Chip Communications. In *Annual Meeting IEEE Lasers and Electro-Optics Society*, Nov 2008.

[20] Johnnie Chan, Gilbert Hendry, Aleksandr Biberman, Keren Bergman, and Luca P. Carloni. PhoenixSim: A Simulator for Physical-Layer Analysis of Chip-Scale Photonic Interconnection Networks. In *Proc. Design, Automation and Test in Eurpoe*, 2010.

[21] M.-C. Frank Chang, Eran Socher, Sai-Wang Tam, Jason Cong, and Glenn Reinman. RF Interconnects for Communications On-Chip. In *Proc. Int. Symp. on Physical Design*, 2008.

[22] Guoqing Chen, Hui Chen, Mikhail Haurylau, Nicholas Nelson, Philippe M. Fauchet, Eby G. Friedman, and David Albonesi. Predictions of CMOS Compatible On-Chip Optical Interconnect. In *Proc. System-Level Interconnect Prediction*, pages 13–20, 2005.

[23] Long Chen, Po Dong, and Michal Lipson. High Performance Germanium Photodetectors Integrated on Submicron Silicon Waveguides by Low Temperature Wafer Bonding. In *Optics Express*, July 2008.

[24] Long Chen, Kyle Preston, Sasikanth Manipatruni, and Michal Lipson. Integrated GHz Silicon Photonic Interconnect with Micrometer-scale Modulators and Detectors. In *Optics Express*, volume 17(17), 2009.

[25] Ray T. Chen. Optical Interconnects and VLSI Photonics. In *IEEE LEOS 2004 Summer Topical Meeting, LEOS Newsletters No.5*, pages 8–9, June 2004.

[26] Ray T. Chen, Lei Lin, Chulchae Choi, Yujie J. Liu, Bipin Bihari, L. Wu, Suning Tang, R. Wickman, B. Picor, M. K. Hibbs-Brenner, J. Bristow, and Y. S. Liu. Fully Embedded Board-Level Guided-Wave Optoelectronic Interconnects. In *Proceedings of the IEEE, Vol. 88(6)*, 2000.

[27] Tai-Chen Chen, Guang-Wan Liao, and Yao-Wen Chang. Predictive Formulae for OPC with Applications to Lithography-Friendly Routing. In *Proc. Design Automation Conf.*, June 2008.

[28] Xiaonan Chen, Yun-Sheng Chen, Yang Zhao, Wei Jiang, and Ray T. Chen. Capacitor-embedded 0.54pJ/bit Silicon-slot Photonic Crystal Waveguide Modulator. In *Optics Letters*, volume 34(5), pages 602–604, 2009.

[29] Hoyeol Cho et al. Modeling of the Performance of Carbon Nanotube Bundle, Cu/Low-K and Optical On-Chip Global Interconnects. In *Proc. System-Level Interconnect Prediction*, March 2007.

[30] Hoyeol Cho, Pawan Kapur, and Krishna C. Saraswat. Power Comparison Between High-Speed Electrical and Optical Interconnect for Interchip Communication. In *Journal of Lightwave Technology*, volume 22, Sept. 2004.

[31] Minsik Cho, Katrina Lu, Kun Yuan, and David Z. Pan. BoxRouter 2.0: Architecture and Implementation of a Hybrid and Robust Global Router. In *Proc. Int. Conf. on Computer Aided Design*, Nov. 2007.

[32] Minsik Cho and David Z. Pan. BoxRouter: A New Global Router Based on Box Expansion. In *IEEE Transactions on Computer-Aided Design*, volume 26(12), pages 2130–2143, 2007.

[33] Minsik Cho, Kun Yuan, Yongchan Ban, and David Z. Pan. ELIAD: Efficient Lithography Aware Detailed Routing Algorithm with Compact and Macro Post-OPC Printability Prediction. In *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 09.

[34] Minsik Cho, Kun Yuan, Yongchan Ban, and David Z. Pan. ELIAD: Efficient Lithography Aware Detailed Router with Compact Printability Prediction. In *Proc. Design Automation Conf.*, June 2008.

[35] J. Cong, T. Kong, and D. Z. Pan. Buffer Block Planning for Interconnect Planning and Prediction. In *IEEE Transactions on VLSI Systems*, volume 9(6), pages 929–937, 2001.

[36] Jason Cong and David Z. Pan. Interconnect Performance Estimation Models for Design Planning. In *IEEE Transactions on Computer-Aided Design*, volume 20(6), pages 739–752, 2001.

[37] Jason Cong and David Z. Pan. Wire Width Planning for Interconnect Performance Optimization. In *IEEE Transactions on Computer-Aided Design*, volume 21(3), pages 319–329, 2002.

[38] William J. Dally and Brian Towles. Route Packets, Not Wires: On-Chip Interconnection Networks. In *Proc. Design Automation Conf.*, 2001.

[39] Duo Ding, Jhih-Rong Gao, Kun Yuan, and David Z. Pan. AENEID: A Generic Lithography-Friendly Detailed Router Based on Post-RET Data Learning and Hotspot Detection. In *Proc. Design Automation Conf.*, 2011.

[40] Duo Ding and David Z. Pan. OIL: A Nano-photonics Optical Interconnect Library for a New Photonic Networks-on-Chip Architecture. In *Proc. System-Level Interconnect Prediction*, 2009.

[41] Duo Ding, J. Andres Torres, and David Z. Pan. High Performance Lithography Hotspot Detection with Successively Refined Pattern Identifications and Machine Learning. In *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2011.

[42] Duo Ding, J. Andres Torres, Fidor G. Pikus, and David Z. Pan. High Performance Lithographic Hotspot Detection Using Hierarchically Re-

fined Machine Learning. In *Proc. Asia and South Pacific Design Automation Conf.*, 2011.

[43] Duo Ding, Xiang Wu, Joydeep Ghosh, and David Z. Pan. Machine Learning based Lithographic Hotspot Detection with Critical Feature Extraction and Classification. In *International Conference for Integrated Circuit Design Technology*, Austin, TX, 2009.

[44] Duo Ding, Yilin Zhang, Haiyu Huang, Ray T. Chen, and David Z. Pan. O-Router: An Optical Routing Framework for Low Power On-Chip Silicon Nano-photonics Integration. In *Proc. Design Automation Conf.*, July 2009.

[45] J. Dong, J. Zhang, and Z. Chen. Neural Network based Algorithm for Multi-Constrained Shortest Path Problem. 2007.

[46] Po Dong, Shirong Liao, Dazeng Feng, Hong Liang, Dawei Zheng, Roshanak Shafiiha, Cheng-Chih Kung, Wei Qian, Guoliang Li, Xuezhe Zheng, Ashok V. Krishnamoorthy, and Mehdi Asghari. Low Vpp, Ultralow-energy, Compact, High-speed Silicon Electro-Optic Modulator. In *OPTICS EXPRESS, Vol.17(25)*, 2009.

[47] D. G. Drmanac, F. Liu, and L.-C. Wang. Predicting Variability in Nanoscale Lithography Processes. In *Proc. Design Automation Conf.*, San Francisco, CA, 2009.

[48] Shaya Fainman. Nanophotonics for On-Chip Integration of WDM Systems. In *DARPA WDM Worshop*, April 2000.

[49] R.-E. Fan, P.-H. Chen, and C.-J. Lin. Working Set Selection Using Second Order Information for Training Support Vector Machines. In *Journal of Machine Learning Research*, 2005.

[50] Joydeep Ghosh. Adaptive and Neural Methods for Image Segmentation. In Bovik [17], pages 401–414.

[51] J. W. Goodman. Optical Interconnects for VLSI Systems. In *Proc. of IEEE*, volume 72, pages 850–866, July 1984.

[52] William M. J. Green, Michael J. Rooks, Lidija Sekaric, and Yurii A. Vlasov. Ultra-compact, Low RF Power, 10Gb/s Silicon Mach-Zehnder Modulator. In *Optics Express*, Dec 2007.

[53] William M. J. Green, Michael J. Rooks, Lidija Sekaric, and Yurii A. Vlasov. Ultra-Compact Low RF Power 10Gb/s Silicon MachZehnder Modulator. In *Proc. of the 20th Annual Meeting of the IEEE Lasers & Electro Optics Society*, 2007.

[54] William M. J. Green, Fengnian Xia, Solomon Assefa, Michael J. Rooks, Lidija Sekaric, and Yurii A. Vlasov. Silicon Photonic Wire Circuits for On-Chip Optical Interconnects. In *Proc. of SPIE*, 2008.

[55] Lanlan Gu, Wei Jiang, Xiaonan Chen, Li Wang, and Ray T. Chen. High Speed Silicon Photonic Crystal Waveguide Modulator for Low Voltage Applicatoin. In *Applied Physics Letters, 90, 071105*, 2007.

[56] Naomi Halas, Peter Nordlander, and Yehia Massoud. Subwavelength Nanophotonics for Future Interconnects and Architectures. In *Semiconductor Research Corp. Annual Review*, 2010.

[57] Gilbert Hendry, Shoaib Kamil, Aleksandr Biberman, Johnnie Chan, Benjamin G. Lee, Marghoob Mohiyuddin, Ankit Jain, Keren Bergman, Luca P. Carloni, John Kubiatowicz, Leonid Oliker, and John Shalf. Analysis of Photonic Networks for a Chip Multi-Processor Using Scientific Applications. In *International Synposium on Networks-on-Chip*, May 2009.

[58] Seng-Tiong Ho, Yingyan Huang, and Jing Ma. InP Photonic Integrated Circuit and DWDM-on-Chip Technology. In *IEEE Fiber-Optics and Photonics Technology Conference*, Oct. 2007.

[59] Andrew Huang, Cary Gunn, Guo-Liang Li, Yi Liang, Sina Mirsaidi, Adithyaram Narasimha, and Thierry Pinguet. A 10Gb/s Photonic Modulator and WDM MUX/DEMUX Integrated with Electronics in 0.13um SOI CMOS. In *IEEE International Solid-State Circuits Conference*, 2006.

[60] Li-Da Huang and Martin D. F. Wong. Optical Proximity Correction(OPC)-Friendly Maze Routing. In *Proc. Design Automation Conf.*, 2004.

[61] Ankit Jain, Shoaib Kamil, Marghoob Mohiyuddin, John Shalf, and John Kubiatowicz. Performance and Energy Comparison of Electrical and Hybrid Photonic Networks of CMPs. In *The Applied Power Electronics Conference and Exposition*, 2008.

[62] Wooyound Jang, Duo Ding, and David Z. Pan. A Voltage Frequency Island Aware Energy Optimization Framework for Networks-on-Chip. In *Proc. Int. Conf. on Computer Aided Design*, 2008.

[63] Wooyoung Jang and David Z. Pan. An SDRAM-Aware Router for Networks-on-Chip. In *Proc. Design Automation Conf.*, July 2009.

[64] Yongqiang Jiang, Wei Jiang, Lanlan Gu, Xiaonan Chen, and Ray T. Chen. 80-micron Interaction Length Silicon Photonic Crystal Waveguide Modulator. In *Applied Physics Letters*, 2005.

[65] Ajay Joshi, Christopher Batten, Yong-Jin Kwon, Scott Beamer, Imran Shamim, Krste Asanovic, and Vladimir Stojanovic. Silicon-Photonic Clos Networks for Global On-Chip Communication. In *International Synposium on Networks-on-Chip*, 2009.

[66] A. B. Kahng, C.-H. Park, and X. Xu. Fast Dual Graph based Hotspot Detection. In *Proc. of SPIE*, volume 6349, 2006.

[67] J. Kim and M. Fan. Hotspot Detection on Post-OPC Layout using Full Chip Simulation based Berification Tool: A Case Study with Aerial Image Simulation. In *Proc. of SPIE*, volume 5256, 2003.

[68] M. J. Kobrinsky. On-Chip Optical Interconnects. In *INTEL Technol. J8(2)*, pages 129–141, 2004.

[69] S. J. Koester, G. Dehlinger, J. D. Schaub, J. O. Chu, Q. C. Ouyang, and A. Grill. Germanium-on-Insulator Photodetectors. In *IEEE International Conference on Group VI Photonics*, pages 171–173, 2005.

[70] S. J. Koester, G. Dehlinger, J. D. Schaub, J. O. Chu, Q. C. Ouyang, and A. Grill. Germanium-on-Insulator Photodetectors. In *IEEE International Conference on Group IV Photonics*, Sept. 2005.

[71] Kyung-Hoae Koo, Pawan Kapur, and Krishna C. Saraswat. Compact Performance Models and Comparisons for Gigascale On-Chip Global Interconnect Technologies. In *IEEE Transactions on Electron Devices*, July 2009.

[72] M. K. Kozlov, S. P. Tarasov, and Leonid G. Khachiyan. Polynomial Solvability of Convex Quadratic Programming. In *Soviet Mathematics - Doklady 20, pp. 1108-1111*, 1979.

[73] Shashi Kumar, Axel Jantsch, Mikael Millberg, Johny Oberg, Juha-Pekka Soininen, Martti Forsell, Kari Tiensyrja, and Ahmed Hemani. A Network on Chip Architecture and Design Methodology. In *Proc. IEEE Annual Symp. on VLSI*, 2002.

[74] Benjamin G. Lee, Aleksandr Biberman, Keren Bergman, Nicolas Sherwood-Droz, and Michal Lipson. Multi-Wavelength Message Routing in a

Non-Blocking Four-Port Bidirectional Switch Fabric for Silicon Photonic Networks-on-Chip. In *Optical Fiber Communication Conference*, March 2009.

[75] Benjamin G. Lee et al. Demonstrated 4X4 Gbps Silicon Photonic Integrated Parallel Electronic to WDM Interface. In *Optical Fiber Communication Conference and Exposition*, May 2007.

[76] B.G. Lee, Xiaogang Chen, Aleksandr Biberman, Xiaoping Liu, I-Wei Hsieh, Cheng-Yun Chou, Jerry I. Dadap, Fengnian Xia, William M. J. Green, Lidija Sekaric, Yurii A. Vlasov, Richard M. Osgood, and Keren Bergman. Ultrahigh-Bandwidth Silicon Photonic Nanowire Waveguides for On-Chip Networks. In *IEEE Photonic Technology Letters*, 2008.

[77] Jian Li. 3D Integration for Computer Architects. In *Proc. Int. Symp. on Computer Architecture*, June 2008.

[78] Zheng Li, Dan Fay, Alan Mickelson, Li Shang, Manish Vachharajani, Dejan Filipovic, Wounjhang Park, and Yihe Sun. Spectrum: A Hybrid Nanophotonic-electric On-chip Network. In *Proc. Design Automation Conf.*, 2009.

[79] Ling Liao, Dean Samara-Rubio, Michael Morse, Ansheng Liu, Dexter Hodge, Doron Rubin, Ulrich Keil, and Thorkild Franck. High Speed Silicon Mach-Zehnder Modulator. In *Optics Express*, April 2005.

[80] Lars Liebmann, Jongwook Kye, Byung-Sung Kim, Lei Yaun, and Jean-Pierre Geronimi. Taming The Final Frontier of Optical Lithography: Design for Sub-resolution Patterning. In *Proc. of SPIE*, 2010.

[81] Ansheng Liu, Ling Liao, Doron Rubin, Hat Nguyen, Berkehan Ciftcioglu, Yoel Chetrit, Nahum Izhaky, and Mario Paniccia. High-Speed Optical Modulation based on Carrier Depletion in a Silicon Waveguide. In *Optics Express*, Jan. 2007.

[82] Tao Luo, David Newmark, and David Z. Pan. DPlace 2.0: A Stable and Efficient Analytical Placement based on Diffusion. In *Proc. Asia and South Pacific Design Automation Conf.*, 2008.

[83] Ning Ma, Justin Ghan, Sandipan Mishra, Costas Spanos, Kameshwar Poolla, Norma Rodriguez, and Luigi Capodieci. Automatic Hotspot Classification using Pattern-based Clustering. In *Proc. of SPIE*, volume 6925, 2007.

[84] Duncan L. MacFarlane, Manasi Peshave, Wei Zhou, Nahid Sultana, Marc P. Christensen, Nathan R. Huntoon, and Gary A. Evans. Four port nanophotonic couplers for dense, planar integrated optics. In *IEEE/LEOS Internationall Conference on Optical MEMs and Nanophotonics*, Aug. 2008.

[85] Linnell Martinez and Michal Lipson. High Confinement Suspended Micro-ring Resonators in Silicon-on-Insulator. In *Optics Express*, June 2006.

[86] David A. B. Miller. Device Requirement for Optical Interconnects to Silicon Chips. In *Proc. of IEEE Special Issue on Silicon Photonics*, 2009.

[87] Jacob R. Minz, Somaskanda Thyagaraja, and Sung Kyu Lim. Optical Routing for 3D System-on-Package. In *Proc. Design, Automation and Test in Eurpoe*, March 2006.

[88] Jacob R. Minz, Somaskanda Thyagaraja, and Sung Kyu Lim. Optical Routing for 3-D System-on-Package. In *IEEE Transactions on Components and Packaging Technologies*, Dec 2007.

[89] J. Mitra, P. Yu, and D. Z. Pan. RADAR: RET-Aware Detailed Routing using Fast Lithography Simulation. In *Proc. Design Automation Conf.*, June 2005.

[90] Dirk Muller. Optimizing Yield in Global Routing. In *Proc. Int. Conf. on Computer Aided Design*, Nov 2006.

[91] Richard Murphy. On the Effects of Memory Latency and Bandwidth on Supercomputer Application Performance. Sept. 2007.

[92] Norimasa Nagase, Kouichi Suzuki, Kazuhiko Takahashi, Masahiko Minemura, Satoshi Yamauchi, and Tomoyuki Okada. Study of Hotspot Detection using Neural Network Judgement. In *Proc. of SPIE*, volume 6607, 2007.

[93] Ian. O'Connor. Optical Solutions for System-Level Interconnect. In *Proc. System-Level Interconnect Prediction*, 2004.

[94] I. O'Conor and Frederic Gaffiot. On-Chip Optical Interconnect for Low-Power. In *E. Macii(Ed), Ultra-Low Power Electronics and Design*, Kulwer, Dordrecht, 2004.

[95] M. Oehme, J. Werner, E. Kasper, M. Jutzi, and M. Berroth. High Bandwidth Ge p-i-n Photodetector Integrated on Si. In *Applied Physics Letter*, 2006.

[96] Ali K. Okyay, Duygu Kuzum, Salman Latif, David A. B. Miller, and Krishna C. Saraswat. Silicon Germanium CMOS Optoelectronic Switching Device: Bringing Light to Latch. In *IEEE Transactions on Electron Devices*, Dec 2007.

[97] David Z. Pan, Minsik Cho, and Kun Yuan. Manufacturability Aware Routing in Nanometer VLSI. In *Foundations and Trends in Electronic Design Automation*, 2010.

[98] Yan Pan, P. Kumar, J. Kim, G. Memik, and A. Choudhary. Firefly: Illuminating Future Network-on-Chip with Nanophotonics. In *Proc. Int. Symp. on Computer Architecture*, 2009.

[99] Yan Pan, Prabhat Kumar, John Kim, Gokhan Memik, Yu Zhang, and Alok Choudhary. Firefly: Illuminating Future Network-on-Chip with Nanophotonics. In *Proc. Int. Symp. on Computer Architecture*, 2009.

[100] Michele Petracca, Keren Bergman, and Luca P. Carloni. Photonic Networks-on-Chip: Opportunities and Challenges. In *Proc. IEEE Int. Symp. on Circuits and Systems*, 2008.

[101] Payam Rabiei, William H. Steier, Cheng Zhang, and Larry R. Dalton. Polymer Micro-Ring Filters and Modulators. In *J. of Lightwave Technology*, 2002.

[102] Martin Riedmiller and Heinrich Braun. A Direct Adaptive Method for Faster Backpropagation Learning: the RPROP Algorithm. In *IEEE Int. Conf. on Neural Networks*, 1993.

[103] Ed Roseboom, Mark Rossman, Fang-Cheng Chang, and Philippe Hurat. Automated Full-Chip Hotspot Detection and Removal Flow for Interconnect Layers of Cell-Based Designs. In *Proc. of SPIE*, volume 6521, 2007.

[104] Subal Sahni, Xi Luo, Jian Liu, Ya-Hong Xie, and Eli Yablonovitch. Junction Field-effect-transistor based Germanium Photodetector on Silicon-on-Insulator. In *Optics Letters*, May 2008.

[105] Krishna Saraswat, Hoyeol Cho, Pawan Kapur, and Kyung-Hoae Koo. Performance Comparison between Copper, Carbon Nanotube and Optical Interconnects. In *Proc. IEEE Int. Symp. on Circuits and Systems*, 2008.

[106] Prashant Saxena, Noel Menezes, Pasquale Cocchini, and Desmond A. Kirkpatrick. Repeator Scaling and its Impact on CAD. In *IEEE Trans. on Computer Aided Design*, volume 23(4), pages 151–163, 2004.

[107] Bradley Schmidt, Qianfan Xu, Jagat Shakya, Sasikanth Manipatruni, and Michal Lipson. Compact Electro-optic Modulator on Silicon-on-Insulator Substrates using Cavities with Ultra-small Modal Volumes. In *Optics Express*, March 2007.

[108] Assaf Shacham, Keren Bergman, and Luca P. Carloni. On the Design of a Photonic Networks-on-Chip. In *International Synposium on Networks-on-Chip*, May 2007.

[109] Assaf Shacham, Keren Bergman, and Luca P. Carloni. The Case for Low Power Photonic Networks-on-Chip. In *Proc. Design Automation Conf.*, Jun 2007.

[110] Assaf Shacham, Keren Bergman, and Luca P. Carloni. Photonic Networks-on-Chip for Future Generations of Chip Multiprocessors. In *IEEE Transactions on Computers*, Sep 2008.

[111] Assaf Shacham, Keren Bergman, and Luca P. Carloni. Photonic Networks-on-Chip for Future Generations of Chip Multiprocessors. In *IEEE Transactions on Computers*, 2008.

[112] Assaf Shacham, Keren Bergman, and Luca P. Carloni. Photonic Networks-on-Chip for Future Generations of Chip Multiprocessors. In *IEEE*

Transaction on Computers, 2008.

[113] Assaf Shacham, Benjamin G. Lee, Aleksandr Biberman, Keren Bergman, and Luca P. Carloni. Photonic NoC for DMA Communications in Chip Multiprocessors. In *IEEE Symposium on High Performance Interconnects*, 2007.

[114] Jagdeep Shah. Ultraperformance Nanophotonic Intrachip Communications: UNIC. In *DARPA/MTO - Frontiers of Extreme Computing*, Oct. 2007.

[115] Steven J. Spector, Theodore M. Lyszczarz, Michael W. Geis, Jung U. Yoon, Donna M. Lennon, and Sandra J. Deneault. Reflections in Silicon on Insulator Waveguides and Ring Resonators. In *Integrated Photonic Research*, 2004.

[116] Navin Srivastava and Kaustav Banerjee. Performance Analysis of Carbon Nanotube Interconnects for VLSI Applications. In *Proc. Int. Conf. on Computer Aided Design*, 2005.

[117] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. Addison-Wesley, 2006.

[118] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. Classification: Alternative Techniques. In *Introduction to Data Mining* [117], pages 207–315.

[119] J. Andres Torres and N. C. Berglund. Integrated Circuit DFM Framework for Deep Sub-Wavelength Processes. In *Proc. SPIE*, 2005.

[120] J. Andres Torres, M. Hofmann, and O. Otto. Directional 2D functions as models for fast layout pattern transfer verification. In *Proc. SPIE*, 2009.

[121] TSMC. Design for Manufacturability. In *TSMC Technology Symposium, Austin Texas*, May 2008.

[122] Kagan Tumer and Joydeep Ghosh. Robust Combining of Disparate Classifiers through Order Statistics. In *Pattern Analysis & Applications, pp. 189-200*, 2002.

[123] Yurii Vlasov. Silicon Photonics for Next Generation Computing Systems. In *European Conference on Optical Communications*, Sep 2008.

[124] Yurii Vlasov, William M. J. Green, and Fengnian Xia. High-throughput Silicon Nanophotonic Wavelength-insensitive Switch for On-Chip Optical Networks. In *nature photonics*, April 2008.

[125] Yurii A. Vlasov, Martin O'Boyle, Hendrik F. Hamann, and Sharee J. McNab. Active Control of Slow Light on a Chip with Photonic Crystal Waveguides. In *Nature*, volume 438, 2005.

[126] Kazumi Wada, Hsin-Chiao Luan, Desmond R. Lim, and Lionel C. Kimerling. On-Chip Interconnection Beyond Semiconductor Roadmap - Silicon Microphotonics. In *Proc. of SPIE*, 2002.

[127] Howard Wang, Michele Petracca, Aleksandr Biberman, Benjamin G. Lee, Luca P. Carloni, and Keren Bergman. Nanophotonic Optical Interconnect Network Architecture for On-Chip and Off-Chip Communications. In *Optical Fiber Communication Conference*, 2008.

[128] Xiaolong Wang, Wei Jiang, Li Wang, Hai Bi, and Ray T. Chen. Fully Embedded Board-Level Optical Interconnect From Waveguide Fabrication to Device Integration. In *Journal of Lightwave Technology*, Jan 2008.

[129] Neil H. E. Weste and David Money Harris. CMOS VLSI Design: A Circuit and Systems Perspective (4th Edition), Addison-Wesley. 2010.

[130] Y.-R. Wu, M.-C. Tsai, and T.-C. Wang. Maze Routing with OPC Consideration. In *Proc. Asia and South Pacific Design Automation Conf.*, 2005.

[131] Jen-Yi Wuu, Fedor G. Pikus, and Malgorzata Marek-Sadowska. Efficient Approach to Early Detection of Lithographic Hotspots Using Machine Learning Systems and Pattern Matching. In *Proc. of SPIE*, 2011.

[132] Jen-Yi Wuu, Fedor G. Pikus, and Malgorzata Marek-Sadowska. Rapid Layout Pattern Classification. In *Proc. Asia and South Pacific Design Automation Conf.*, 2011.

[133] Jen-Yi Wuu, Fedor G. Pikus, J. Andres Torres, and Malgorzata Marek-Sadowska. Detecting Context Sensitive Hot Spots in Standard Cell

Libraries. In *Proc. of SPIE*, 2009.

[134] Fengnian Xia, Lidija Sekaric, and Yurii A. Vlasov. Ultracompact Optical Buffers on A Silicon Chip. In *Nature Photonics*, Jan 2007.

[135] Jingyu Xu, Subarna Sinha, and Charles C. Chiang. Accurate Detection for Process Hotspots with Vias and Incomplete Specification. In *Proc. Int. Conf. on Computer Aided Design*, 2007.

[136] Qianfan Xu, Sasikanth Manipatruni, Brad Schmidt, Jagat Shakya, and Michal Lipson. 12.5 Gbit/s Carrier-Injection based Silicon Micro-ring Silicon Modulators. Jan 2007.

[137] F. Yang, Y. Cai, Q. Zhou, and J. Hu. SAT Based Multi-Net Rip-up-and-Reroute for Manufacturing Hotspot Removal. In *Proc. Design, Automation and Test in Eurpoe*, 2010.

[138] H. Yao, S. Sinha, C. Chiang, X. Hong, and Y. Cai. Efficient Process Hotspot Detection using Range Pattern Matching. In *Proc. Int. Conf. on Computer Aided Design*, 2006.

[139] Tao Yin, Rami Cohen, Mike M. Morse, Gadi Sarid, Yoel Chetrit, Doron Rubin, and Mario J. Paniccia. 31GHz Ge n-i-p Waveguide Photodetectors on Silicon-on-Insulator Substrate. In *Optics Express*, Oct. 2007.

[140] Ian A. Young, Edris Mohammed, Jason T. S. Liao, Alexandra M. Kern, Samuel Palermo, Bruce A. Block, Miriam R. Reshotko, and Peter L. D.

Chang. Optical I/O Technology for Tera-Scale Computing. In *IEEE Journal of Solid-state Circuits, Vol.45, No.1*, 2010.

[141] Peng Yu and David Z. Pan. TIP-OPC: A New Topological Invariant Paradigm for Pixed based Optical Proximity Correction. In *Proc. Int. Conf. on Computer Aided Design*, 2007.

[142] Yanheng Zhang, Yue Xu, and Chris Chu. FastRoute 3.0: A Fast and High Quality Global Router Based on Virtual Capacity. In *Proc. Int. Conf. on Computer Aided Design*, 2008.

Index

Vita

Duo Ding was born in Jilin (literally "The Woods of Luck") City, Jilin Province, P. R. China on August 6, 1983. He received a Bachelor degree of Science from Harbin Institute of Technology in 2006. In 2008, he received a degree of Master of Science in Engineering from The University of Texas at Austin, majoring in Signal Processing and Computer Networks. In the summer of 2008, Duo joined the UT Design Automation Lab at the Computer Engineering Research Center. Duo joined Oracle (former SUN Microsystems) Microelectronics Center in August 2011, after he obtained his Ph.D. degree in Electrical and Computer Engineering.

Duo's main research involves various VLSI Computer-Aided Design methodologies and optimization algorithms for efficient IC design and manufacturing closure. In the summer of 2009, Duo joined Mentor Graphics Corp. at Wilsonville, OR, where he researched and developed a high performance engine for detecting lithography process hotspots. His contribution in this work is patented as a part of CALIBRE product, the leading commercial software in lithography simulation. Duo also works on device modeling and low power physical design of optical electronic integrated circuits.

The impacts of Duo's works are recognized with Best Student Paper Award and Best Paper Award nomination at prestigious IEEE/ACM interna-

tional conferences, as well as summer school fellowships and travel grants at IEEE/ACM Design Automation Conference 2009-2011, IEEE SLIP'09, IEEE CANDE'09. Duo enjoys playing guitar in his spare time. He also enjoys ancient Chinese and Greek philosophy.

Journal Articles

J4. **Duo Ding**, J. Andres Torres and David Z. Pan, "High Performance Lithography Hotspot Detection with Successively Refined Pattern Identifications and Machine Learning", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2011.*

J3. Ryan A. Integlia, Lianghong Yin, **Duo Ding**, David Z. Pan, Douglas M. Gill and Wei Jiang, "Parallel-Coupled Dual Racetrack Silicon Microresonators for Quadrature Amplitude Modulation", *Optics Express, Vol.19, pp14892-14902, 2011.*

J2. Xinyuan Dou, Xiaolong Wang, Xiaohui Lin, **Duo Ding**, David Z. Pan and Ray T. Chen, "Highly Flexible Polymeric Optical Waveguide for Out-of-Plane Optical Interconnects", *Optics Express, Vol.18, pp16227-16233, 2010.*

J1. Xinyuan Dou, Xiaolong Wang, Haiyu Huang, Xiaohui Lin, **Duo Ding**, David Z. Pan and Ray T. Chen, "Polymeric Waveguides with Embedded Micro-mirrors Formed by Metallic Hard Mold", *Optics Express, Vol.18, pp378-385, 2009.*

Conference Papers

C14. Duo Ding, Bei Yu, Joydeep Ghosh and David Z. Pan, "EPIC: Efficient Prediction of IC Manufacturing Hotspots With A Unified Meta-Classification Formulation", *submitted to IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC), 2012.*

C13. Duo Ding, Bei Yu and David Z. Pan, "GLOW: A Global Router for Low-Power Thermal-reliable Interconnect Synthesis Using Photonic Wavelength Multiplexing", *submitted to IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC), 2012.*

C12. Bei Yu, Kun Yuan, Boyang Zhang, **Duo Ding** and David Z. Pan, "Layout Decomposition for Triple Patterning Lithography", *IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2011* (**nominated as Best Paper Award candidate**).

C11. Duo Ding, "VLSI CAD for Nanometer IC Manufacturability, Yield and Physical Verification", *IEEE/ACM Design Automation Conference (DAC) SIGDA Ph.D. Forum, San Diego, June 2011.*

C10. Duo Ding, Jhih-Rong Gao, Kun Yuan and David Z. Pan, "AENEID: A Generic Lithography-Friendly Detailed Router Based on Post-RET Data Learning and Hotspot Detection", *IEEE/ACM Design Automation Conference (DAC), 2011.*

C9. **Duo Ding**, J. Andres Torres, Fedor G. Pikus and David Z. Pan, "High Performance Lithographic Hotspot Detection using Hierarchically Refined Machine Learning", *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC), 2011.*

C8. **Duo Ding**, J. Andres Torres, Fedor G. Pikus and David Z. Pan, "High Performance Lithography Hotspot Detection with Hierarchically Refined Machine Learning Methods", *IEEE International Workshop on Design for Manufacturability and Yield (DFM&Y), 2010.*

C7. Wooyoung Jang, **Duo Ding** and David Z. Pan, "Voltage and Frequency Island Optimizations for Many-core/Networks-on-Chip Designs", *The 1st International Conference on Green Circuits and Systems (ICGCS), 2010.*

C6. **Duo Ding** and David Z. Pan, "Computer-Aided Design for Low Power High Performance Synthesis for Silicon Nanophotonics On-chip Integration", *IEEE Computer-Aided Network Design Workshop (CANDE), 2009.*

C5. **Duo Ding** and David Z. Pan, "Machine Learning Classification for Process Hotspot Detection with Edge-based Critical Signature Extraction", *Austin Conference on Integrated Systems and Circuits, 2009.*

C4. **Duo Ding** and David Z. Pan, "OIL: A Nanophotonic Optical Interconnect Library for a New Photonic Networks-on-Chip Architecture",

IEEE International Workshop on System Level Interconnect Prediction (SLIP), 2009.

C3. **Duo Ding**, Yilin Zhang, Haiyu Huang, Ray T. Chen and David Z. Pan, "O-Router: An Optical Routing Framework for Low Power On-Chip Silicon Nano-Photonic Integration", *IEEE/ACM Design Automation Conference (DAC) 2009.*

C2. **Duo Ding**, Xiang Wu, Joydeep Ghosh, and David Z. Pan, "Machine Learning based Lithographic Hotspot Detection with Critical Feature Extraction and Classification", *IEEE International Conference on IC Design and Technology (ICICDT), 2009* (awarded as the only **Best Student Paper Award**).

C1. Wooyoung Jang, **Duo Ding** and David Z. Pan, "A Voltage-Frequency Island Aware Energy Optimization Framework for Networks-on-Chip", *IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2008.*

Patents

P1. Juan Andres Torres Robles, Salma Mostafa Fahmy, Peter Louiz Rezk Beshay, Kareem Madkour, Fedor G. Pikus, Jen-Yi Wuu and **Duo Ding**, "Hybrid Hotspot Detection", *U.S. patent application No. 13/191,436.*

Permanent address: dingduo1983@gmail.com

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.